



UNIVERSIDAD
DE MÁLAGA



E.T.S.
INGENIERÍA
INFORMÁTICA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA GRADUADA/O EN INGENIERÍA DEL SOFTWARE

Aplicación web para novelistas

Desarrollo de un perfil y útiles para la creación novelística

Web application to novelists

Development of profile and tools to novelist

Realizado por
Jesús Galán Pozo

Tutorizado por
Luis Manuel Llopís Torres

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, JUNIO DE 2019

Fecha defensa: de julio de 2019

Fdo. El/la Secretario/a del Tribunal



UNIVERSIDAD
DE MÁLAGA



E.T.S.
INGENIERÍA
INFORMÁTICA

Resumen

Aplicación web para novelistas es una plataforma online sencilla, intuitiva y centrada en las mismas/os escritoras/es, que son las/os protagonistas. Esta plataforma ofrece unos servicios a las usuarias/os para organizar, potenciar y perfeccionar su proceso creativo al máximo. Un escritor/a podrá organizar y planificar su novela haciendo uso de un organizador de tareas para tener claro los pasos a seguir, un diccionario y un glosario asociado a cada libro. Podrá potenciar su creatividad y desarrollo de las obras literarias con un proceso de creación de hojas de personajes donde moldeará a los personajes que compondrán la/s novela/s. Para mejora de este proceso y un mayor aprovechamiento del tiempo que el/la escritor/a dedica tendrá a mano una escritura en línea donde, podrá escribir los capítulos que compondrán sus obras. Además, tendrá a mano útiles que serán de ayuda en la fase de corrección del capítulo/os donde recibirá una retroalimentación de análisis ortográfico y análisis de sentimientos.

Puesto que la aplicación es el resultado de un proyecto en grupo, este documento en concreto se centra en el desarrollo de la biblioteca de la escritora/or desde donde el/la escritor/a pueda acceder a todas las funcionalidades. También se cubren funcionalidades como el propio inicio de sesión, cierre de sesión de los usuarios, planificador, glosario de cada libro, diccionario de la escritora/or, registro de usuario, diseño de hojas de personajes y análisis de sentimientos de las obras. A lo largo del mismo, se refleja cada una de las fases en las que ha consistido este trabajo, la metodología empleada para su desarrollo, así como las tecnologías, recursos utilizados y la experiencia que se ha tenido en su correspondiente uso.

Palabras clave: Análisis de sentimientos, diseño de personajes, biblioteca de usuario, planificador y diccionarios con Angular 5, .Net Core 2.1 y CosmosDB de Azure

Abstract

Web application for novelists is a simple, intuitive online platform focused on the writers themselves, who are the main protagonists. This platform offers services to users to organize, enhance and perfect their creative process to the maximum. A writer can organize and plan his novel using a task organizer to have clear steps to follow, a dictionary and a glossary associated with each book. You will be able to boost your creativity and development of literary works with a process of creating character sheets where you will mold the characters who will compose the novels. In order to improve this process and make better use of the time that the writer spends, he will have at hand an online writing system where he will be able to write the chapters that will compose his works. In addition, you will have at hand tools that will be of help in the correction phase of the chapter where you will receive a feedback of orthographic analysis and analysis of feelings.

Since the application is the result of a group project, this particular document focuses on the development of the writer's library from which the writer can access all functionalities. Also covered are functionalities such as the user's own login, user logout, planner, glossary of each book, writer's dictionary, user registration, character sheet design and sentiment analysis of the works. Throughout it, each of the phases in which this work has consisted is reflected, the methodology used for its development, as well as the technologies, resources used and the experience that has been had in its corresponding use.

Keywords: Sentiments analysis, character sheet design, planner, user profile and dictionary with Angular 5, .Net Core 2.1 and CosmosDB in Azure.

Índice

Resumen	1
Abstract	1
Índice	1
Introducción.....	1
Motivación	2
Objetivos.....	3
Estructura de la memoria	4
Especificación de requisitos y casos de uso.	5
RF01 CU01 – Un usuario podrá registrarse en la aplicación	9
RF02 CU02 - El escritor podrá iniciar sesión	12
RF03 CU03 - El escritor podrá cerrar sesión	14
RF04 CU04 - El escritor tendrá acceso a un perfil propio	16
RF05 CU05 - El escritor tendrá acceso a un diccionario personal	18
RF06 - El escritor podrá crear, editar y borrar palabras en su diccionario personal.	20
▪ RF06.1 CU06.1 – El escritor podrá crear palabras en su diccionario personal	20
▪ RF06.2 CU06.2 – El escritor podrá modificar palabras en su diccionario personal.	22
▪ RF06.3 CU06.3 – El escritor podrá eliminar palabras en su diccionario personal...	24
RF08 CU08 - El escritor tendrá acceso a un glosario asociado a cada libro.....	26
RF09 - El escritor podrá crear, editar y borrar palabras en sus glosarios	28
▪ RF09.1 CU09.1 - El escritor podrá crear palabras en sus glosarios	28
▪ RF09.2 CU09.2 - El escritor podrá modificar palabras en sus glosarios.....	30
▪ RF09.3 CU09.3 - El escritor podrá eliminar palabras en sus glosarios.....	32
RF10 CU10 - El escritor tendrá acceso a notas asociadas a cada libro	34
RF11 - El escritor podrá crear, editar y borrar notas en su lista de notas	36
▪ RF11.1 CU11.1 - El escritor podrá crear notas en su lista de notas.....	36
▪ RF11.2 CU11.2 - El escritor podrá modificar notas de su lista de notas.....	38
▪ RF11.3 CU11.3 - El escritor podrá eliminar notas de su lista de notas.....	40

RF12 CU12 - El escritor tendrá acceso a hojas de personajes asociadas a sus libros	42
RF13 - El escritor podrá crear, editar y borrar sus hojas de personajes	44
▪ RF13.1 CU13.1- El escritor podrá crear hojas de personajes.....	44
▪ RF13.2 CU13.2 - El escritor podrá modificar las hojas de personajes	46
▪ RF13.3 CU13.3 - El escritor podrá eliminar las hojas de personajes	48
RF24 CU24 - El escritor podrá acceder a una lista de tareas para planificarse	50
RF25 - El escritor podrá crear, editar y borrar tareas en su lista de tareas	52
▪ RF25.1 CU25.1 - El escritor podrá crear tareas en su lista de tareas	52
▪ RF25.2 CU25.2 - El escritor podrá modificar las tareas de su lista de tareas.....	54
▪ RF25.3 CU25.3 - El escritor podrá eliminar las tareas de su lista de tareas.....	56
RF27 CU27 - El escritor dispondrá de análisis de sentimientos de sus obras escritas.	58
Metodología	63
Elección de tecnologías	67
Azure DevOps.....	68
Azure CosmosDB⁴.....	69
Swagger⁷	70
ASP.Net Core Web Api	70
Visual Studio Code¹⁴	72
Angular¹⁵	72
Diseño del modelo de datos	75
- DictionaryWord	76
- Writer	77
- TaskWriter	77
- Book	78
- GlossaryWord	78
- Note	79
- SentimentsAnalytic	79
- Chapter.....	79
- ScreenPlayEvent	80
- TextAnalytic.....	80
- Character	81
Desarrollo de la aplicación	82
Registro del usuario en la aplicación	83

Inicio de sesión del escritor	86
Cierre de sesión del escritor	90
Perfil del escritor	91
El escritor tendrá un diccionario personal	92
Cada libro tiene un glosario asociado.....	95
Cada libro tiene un listado de notas asociado	98
El escritor tendrá una galería de personajes	101
El escritor podrá acceder a un planificador de tareas.....	104
El escritor podrá tener un análisis de sentimientos de sus obras	109
Pruebas de la aplicación	113
Pruebas unitarias	114
Test de usabilidad	117
Posibles extensiones.....	119
Conclusión	121
Referencias	125
Anexo: Guía de uso para usuario	128
Acceso a la aplicación.....	129
Cerrar sesión	131
Editar datos del perfil.....	131
Planificación de tareas	133
Crear, borrar editar palabra en el diccionario	137
Crear y borrar libro.....	140
Editar detalles de libro	142
Crear, editar y borrar palabra glosario	143
Crear editar y borrar nota	144
Crear y borrar capítulos.....	145
Escribir Capítulo	146
Crear Guion Rápido	148
Corrección de ortografía	150
Exportación de libro	151
Gráfica de análisis de ortografía.....	152
Análisis de Sentimientos	155
Diseño de personajes	156

1

Introducción

Motivación

En todo momento las/os escritoras/es (en adelante se englobará en escritor/escritores) manejan una serie de información que va más allá de lo que el lector final ve a simple vista en las obras que se crean. Toda esa sabiduría e información que han tenido que obtener a través de diferentes fuentes, métodos y tiempo, han tenido que transformarla usando la imaginación y creatividad a su máximo esplendor.

Todo ese proceso inicial de obtención de ideas e información útil donde después el escritor transforma y plasma ese conocimiento asimilado, creando su propio mundo, ya sea de manera digital o en papel. No es en dicha etapa de búsqueda de conocimiento donde se ha querido centrar este trabajo sino en los siguientes pasos que le siguen, cuando el escritor lleva a cabo la acción de estar frente a una pantalla/papel en blanco y empieza desarrollar su mundo e idea.

Este proyecto surge para impulsar el desarrollo de las obras, englobando funcionalidades que permitan al escritor focalizarse en la escritura y reunir toda la información referente al seguimiento de su obra en un único formato; sin que tenga que recurrir a tantas fuentes externas. Con esto se quiere que el proceso de escritura sea lo más eficiente, reduciendo el número de distracciones y brindando funcionalidades muy recurridas por el propio escritor sin tener que navegar a otras webs para consultar cierta información. Además, se ha querido mejorar la experiencia de crear una obra, no solo centrándose en escribir, sino ofreciéndole al escritor otras herramientas para que su creatividad, organización y rendimiento aumenten siendo así la experiencia de escribir un libro más placentera y adherente.

Objetivos

Esta aplicación tiene como objetivos principales mejorar la creación, escritura y desarrollo de novelas, pudiendo acceder de manera sencilla y rápida tan solo desde un navegador web y una conexión a internet.

Se han detectado ciertas etapas en el desarrollo de una obra por las que pasan los escritores de dichas obras y este proyecto se ha querido centrar en las etapas en las que el escritor ya ha buscado información de como montar una novela y pasa a la acción, en esa etapa es en la que se centra este trabajo. En las etapas en las que el escritor pasa a la acción de escribir la novela necesitará una serie de herramientas que le ayudarán a que el proceso sea más liviano y favorable. Como objetivos principales que tiene este proyecto es ofrecer al escritor una escritura en línea cómoda donde pueda dar formato al texto, subrayar, poner en negrita e incluso cambiar el tipo de letra. A esta escritura en línea le siguen herramientas como una biblioteca donde el escritor podrá organizar sus obras y acceder a ellas sin temor a perderlas, un organizador de tareas para poder planificar el proceso en todo momento, una galería de personajes los cuales forman y dan vida a las obras y una serie de análisis que ayudarán a que la obra transmita lo que realmente el escritor desea. También en todo momento mientras escribe podrá programar una guía de los sucesos importantes que deben ocurrir en los capítulos y que no se olvide nada. Todos estos objetivos ayudan al escritor para que pueda plasmar de forma más clara y específica sus ideas sobre cómo va a ser el mundo y los personajes que dan vida a la obra, ahorrando tiempo y esfuerzo buscando en otras webs. Así solo tendrá que enfocarse en desarrollar su historia.

Estructura de la memoria

Para explicar todas las fases por las que se ha pasado para desarrollar este proyecto se va a seguir el guion siguiente, donde se explicarán cada una de las fases y en qué ha consistido cada una de ellas:

1. Especificación de requisitos y casos de uso.
2. Planteamiento y preparación de metodología a seguir.
3. Elección y preparación de tecnologías y entornos con los que desarrollar el proyecto.
4. Desarrollo e implementación del modelo de datos.
5. Desarrollo de la aplicación web: implementación por requisitos de manera transversal.
6. Pruebas de la aplicación.

2

Especificación de requisitos y casos de uso

La aplicación en cuestión en pro de conseguir los objetivos propuestos anteriormente, se definen los siguientes requisitos funcionales.

Primero se van a especificar los requisitos funcionales en una tabla, tabla 1, los cuales harán que la aplicación cumpla los objetivos propuestos. En la tabla se puede observar los requisitos funcionales a desarrollar en la primera columna, en la segunda columna marcados con un ✓, los requisitos que son tema de este desarrollo y por último el código asociado al requisito.

Requisito	Tratados en este documento	Código
Un usuario podrá registrarse en la aplicación	✓	RF01
El escritor podrá iniciar sesión	✓	RF02
El escritor podrá cerrar sesión	✓	RF03
El escritor tendrá acceso a un perfil propio	✓	RF04
El escritor tendrá acceso a un diccionario personal	✓	RF05
El escritor podrá crear, editar y borrar palabras en su diccionario personal.	✓	RF06
El escritor tendrá acceso a una lista de sus libros		RF07
El escritor tendrá acceso a un glosario asociado a cada libro	✓	RF08
El escritor podrá crear, editar y borrar palabras en sus glosarios	✓	RF09
El escritor tendrá acceso a notas asociadas a cada libro	✓	RF10
El escritor podrá crear, editar y borrar notas en su lista de notas	✓	RF11
El escritor tendrá acceso a hojas de personajes asociadas a sus libros	✓	RF12
El escritor podrá crear, editar y borrar sus hojas de personajes	✓	RF13

El escritor podrá crear, editar y borrar un libro		RF14
El escritor podrá acceder a un libro creado y ver sus capítulos		RF15
El escritor podrá crear y borrar capítulos de un libro		RF16
El escritor tendrá acceso a una pantalla de escritura en línea.		RF17
El escritor podrá editar o escribir capítulos en un editor de texto.		RF18
El escritor podrá elegir opciones de tamaños, estilos y alineación en el editor de texto.		RF19
El escritor podrá exportar su libro a formato Word.		RF20
El escritor podrá hacer una corrección de ortografía de sus capítulos.		RF21
El escritor podrá visualizar en una gráfica el análisis de ortografía de un libro completo.		RF22
El escritor dispondrá de un guion rápido en cada capítulo.		RF23
El escritor podrá acceder a una lista de tareas para planificarse.	✓	RF24
El escritor podrá crear, editar y borrar tareas en su lista de tareas	✓	RF25
El escritor podrá subir y visualizar imágenes de su portada, perfil, y personajes.		RF26
El escritor podrá visualizar en una gráfica el análisis de sentimientos de un libro completo.	✓	RF27

Tabla 1. Tabla sobre los requisitos funcionales.

Para un mayor entendimiento y aclaración sobre qué requisitos se van a desarrollar se pasa a explicar de forma más detallada y de manera individual cada uno de los requisitos marcados con un ✓ en la tabla 1.

Para la especificación individual de cada requisito funcional se utilizará la tabla 2 como guía para especificar el requisito funcional. Además, se complementará

esta especificación con el caso de uso asociado, se hará uso de la tabla 3, donde se explicarán los pasos a seguir por el usuario para saber si el caso de uso es correcto. Así se podría verificar que el requisito funcional ha sido cubierto y el objetivo cumplido. Se ha decidido documentar este punto adjuntando diagramas para los casos de uso realizados con la herramienta MagicDraw.

ID	<i>[nombre descriptivo]</i>
Dependencias	<ul style="list-style-type: none"> ▪ <i>[requisitos generales de los que depende]</i> ▪ <i>[otros requisitos de los que depende]</i> ▪ ...
Descripción	Explicación sobre el requisito.
Datos específicos	<ul style="list-style-type: none"> ▪ <i>[datos específicos sobre el concepto relevante]</i> ▪ ...
Prioridad	<i>[prioridad del requisito para la gestión del proyecto]</i>
Comentarios	<i>[observaciones adicionales sobre el requisito]</i>

Tabla 2. Tabla guía para los requisitos funcionales

Código	xx	
Requisito asociado	Nombre del requisito descriptivo	
Objetivo asociado	Nombre del objetivo descriptivo	
Descripción	El caso de uso a probar...	
Precondición	El sistema está en el estado y	
Secuencia normal	Paso	Acción
	1	Acción 1
	2	Acción 2
	3	Acción 3
Postcondición	El sistema está en el estado p	
Excepciones	Paso	Acción
	1	Acción 1
	2	Acción 2
	3	Acción 3
Importancia	Alta - Media - Baja	

Tabla 3 – Tabla guía para los casos de uso.

RF01 | CU01 – Un usuario podrá registrarse en la aplicación

RF01	Un usuario podrá registrarse en la aplicación
Dependencias	<i>No tiene dependencias</i>
Descripción	Los usuarios que visiten la aplicación y no estén aun registrados en la plataforma podrán acceder a un formulario de registro donde introducir sus datos principales para darse de alta en la aplicación y poder acceder a través del inicio de sesión a todas las funcionalidades de la plataforma.
Datos específicos	<ul style="list-style-type: none"> ▪ <i>Dar de alta a usuarios en el sistema a través de un formulario web.</i> ▪ <i>El formulario debe contener los campos de la tabla 5.</i>
Prioridad	<i>Alta</i>
Comentarios	No tiene comentarios

Tabla 4 – Requisito funcional RF01

Campo	Tipo	Restricciones
Nombre completo del usuario.	Texto	-Tamaño mínimo de 3 caracteres.
Email del usuario con la que accederá a la plataforma.	Texto - Email	-Debe cumplir el formato de un email xxx@domino.xxx
Contraseña con la que accederá a la plataforma.	Texto - Password	-Tamaño mínimo de 5 caracteres. -Solo puede estar compuesta por letras mayúsculas, minúsculas y números.
Fecha de nacimiento del usuario.	Date - Datepicker	-Elegir una fecha del calendario.
País de origen.	Selección de una opción	-Elegir una opción del listado.
Ciudad de origen.	Texto	
Alias para la plataforma.	Texto	-Tamaño mínimo de 2 caracteres.
Tiempo que el usuario lleva escribiendo.	Selección de una opción	-Elegir una opción del listado.
Categoría favorita	Selección de una opción	-Elegir una opción del listado.

Tabla 5 – Campos para el formulario de registro.

Código	CU01	
Requisito asociado	Registro de usuarios.	
Objetivo asociado	Registro de usuarios.	
Descripción	El usuario que no esté dado de alta en el sistema podrá hacerlo para poder acceder a la plataforma.	
Precondición	El usuario no está registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	El usuario solicita registrarse en la aplicación.
	2	El sistema proporciona un formulario donde el usuario deberá rellenar los campos pedidos para su posible registro en el sistema.
	3	El usuario introduce los campos pedidos y solicita su registro en la aplicación.
	4	El sistema procesa la petición y almacena los datos del nuevo usuario.
	5	Indica al usuario el éxito/fracaso del proceso.
Postcondición	El usuario está dado de alta en el sistema.	
Excepciones	Paso	Acción
	3	El usuario no rellena los campos obligatorios.
	4	El sistema ya contiene un usuario con los mismos datos.
Importancia	Alta	

Tabla 6 – Caso de uso CU01

RF02 | CU02 - El escritor podrá iniciar sesión

RF02	El escritor podrá iniciar sesión
Dependencias	<i>No tiene dependencias</i>
Descripción	Los usuarios que visiten la aplicación y estén registrados en la plataforma podrán acceder a un formulario de inicio de sesión donde introducirán sus credenciales, correo electrónico con el que se registraron y contraseña, para poder acceder a la plataforma.
Datos específicos	<ul style="list-style-type: none">▪ <i>Los usuarios registrados podrán iniciar sesión a través de un formulario donde tendrán que rellenar 2 campos:</i><ul style="list-style-type: none">-Email con el que se registró-Contraseña
Prioridad	<i>Alta</i>
Comentarios	No tiene comentarios

Tabla 7 – Requisito funcional RF02

Código	CU02	
Requisito asociado	Inicio de sesión.	
Objetivo asociado	Inicio de sesión de los escritores.	
Descripción	El usuario que este dado de alta en el sistema podrá acceder a la plataforma validando sus credenciales a través del formulario de inicio de sesión.	
Precondición	El usuario debe de estar registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	El usuario solicita iniciar sesión en la aplicación.
	2	El sistema proporciona un formulario donde el usuario deberá rellenar los campos pedidos, correo electrónico con el que se registró y contraseña.
	3	El usuario introduce los campos pedidos y solicita su registro en la aplicación.
	4	El sistema procesa la petición
	5	Indica al usuario el éxito/fracaso del proceso.
	5a	Si el paso 5 ha sido un éxito, accederá al perfil del escritor.
	5b	Si el paso 5 ha sido erróneo se informará al usuario con dicho error.
Postcondición	El usuario ha entrado en su perfil.	
Excepciones	Paso	Acción
	3	El usuario no rellena los campos obligatorios.
	4	El sistema no contiene un usuario con los mismos datos.
Importancia	Alta	

Tabla 8 – Caso de uso CU02

RF03 | CU03 - El escritor podrá cerrar sesión

RF03	El escritor podrá cerrar sesión
Dependencias	RF02
Descripción	Los escritores que hayan iniciado sesión deben de poder cerrar sesión en el momento que decidan.
Datos específicos	<ul style="list-style-type: none">▪ Los escritores podrán cerrar sesión.
Prioridad	Alta
Comentarios	No tiene comentarios

Tabla 9 – Requisito funcional RF03

Código	CU03	
Requisito asociado	Cerrar sesión.	
Objetivo asociado	Cierre de sesión de los escritores.	
Descripción	El escritor que haya iniciado sesión podrá cerrar su sesión.	
Precondición	El escritor debe haber iniciado sesión en el sistema.	
Secuencia normal	Paso	Acción
	1	El escritor localizará un botón para el cierre de sesión en la barra superior de navegación.
	2	El escritor pulsará dicho botón con el texto de “cerrar sesión”.
	3	La aplicación se encargará de borrar los datos de la sesión del usuario, mostrando el contenido para los usuarios.
	4	El escritor no podrá acceder a ningún enlace como “mi perfil”.
	5	La aplicación redirigirá a la página de inicio.
Postcondición	El escritor ya no tiene la sesión activa.	
Excepciones	Paso	Acción
	3	La aplicación no borra correctamente datos de sesión del usuario que hace que no se cierre correctamente la sesión.
	5	El sistema no redirige a la página correctamente.
Importancia	Media	

Tabla 10 – Caso de uso CU03

RF04 | CU04 - El escritor tendrá acceso a un perfil propio

RF04	El escritor tendrá acceso a un perfil propio
Dependencias	<i>Sin dependencias.</i>
Descripción	Los escritores tendrán acceso a un perfil donde será el núcleo de la plataforma concentrando los enlaces/accesos a las herramientas principales y de fácil acceso.
Datos específicos	<ul style="list-style-type: none">▪ <i>Tendrá información básica del escritor.</i>▪ <i>Tendrá una lista con los libros y un pequeño resumen donde pueda ver toda su biblioteca de un solo vistazo.</i>
Prioridad	<i>Alta</i>
Comentarios	No tiene comentarios

Tabla 11 – Requisito funcional RF04

Código	CU04	
Requisito asociado	El escritor tendrá acceso a un perfil propio.	
Objetivo asociado	El escritor tendrá un perfil propio donde convergerán gran parte de las herramientas.	
Descripción	El perfil del escritor tiene la información principal del propio escritor y toda la biblioteca de obras que lleva escrita.	
Precondición	El escritor debe haber iniciado sesión en el sistema o acceder a través del enlace de la barra de navegación.	
Secuencia normal	Paso	Acción
	1	El escritor podrá ver de un solo vistazo su propia información como escritor.
	2	El escritor podrá ver a simple vista una sección con todos los libros que lleva escrito y un pequeño resumen.
	3	Podrá acceder a los libros pinchando sobre el elemento.
Postcondición	El escritor puede acceder a los libros que está escribiendo.	
Excepciones	Paso	Acción
	1,2,3	Fallo en la sincronización con la base de datos que provoca que los datos no se carguen correctamente.
Importancia	Alta	

Tabla 12 – Caso de uso CU04

RF05 | CU05 - El escritor tendrá acceso a un diccionario personal

RF05	El escritor tendrá acceso a un diccionario personal
Dependencias	<i>RF04</i>
Descripción	Los escritores tendrán acceso diccionario personal global donde podrán ir almacenando palabras que encuentren interesantes y útiles para posibles usos futuros en sus propias obras.
Datos específicos	<i>Sin datos específicos.</i>
Prioridad	Media
Comentarios	No tiene comentarios

Tabla 13 – Requisito funcional RF05

Código	CU05	
Requisito asociado	El escritor tendrá acceso a un diccionario personal.	
Objetivo asociado	El escritor tendrá acceso a un diccionario personal.	
Descripción	El escritor tendrá la posibilidad de consultar su diccionario personal para ver las palabras guardadas para su posible uso futuro en obras propias.	
Precondición	El escritor debe acceder al perfil para poder entrar a su propio diccionario personal.	
Secuencia normal	Paso	Acción
	1	El escritor podrá ver de un solo vistazo todas las palabras que ha guardado en su diccionario personal.
	2	El escritor podrá pinchar sobre la palabra para que se despliegue la descripción de dicha palabra y obtener más información.
	2a	Podrá pinchar de nuevo para cerrar dicha información.
Postcondición	El escritor puede acceder a su diccionario personal de palabras.	
Excepciones	Paso	Acción
	1,2	Fallo en la sincronización con la base de datos que provoca que los datos no se carguen correctamente.
Importancia	Media	

Tabla 14 – Caso de uso CU05

RF06 - El escritor podrá crear, editar y borrar palabras en su diccionario personal.

- **RF06.1 | CU06.1 – El escritor podrá crear palabras en su diccionario personal**

RF06.1	El escritor podrá crear palabras en su diccionario personal.
Dependencias	<i>RF04, RF05</i>
Descripción	Los escritores tendrán acceso diccionario personal global donde podrán ir almacenando palabras que encuentren interesantes y útiles para posibles usos futuros en sus propias obras. Dichas palabras las podrá añadir el propio escritor para su posterior uso.
Datos específicos	<ul style="list-style-type: none">- <i>El diccionario está compuesto por palabras y cada palabra podrá tener una descripción aclarando su significado.</i>
Prioridad	Media
Comentarios	No tiene comentarios

Tabla 15 – Requisito funcional RF06.1

Código	CU06.1	
Requisito asociado	El escritor podrá crear palabras en su diccionario personal.	
Objetivo asociado	El escritor podrá crear palabras en su diccionario personal.	
Descripción	El escritor tendrá la posibilidad de gestionar su diccionario personal para crear todas las palabras que necesite en su diccionario personal.	
Precondición	El escritor debe acceder al perfil para poder entrar a su propio diccionario personal.	
Secuencia normal	Paso	Acción
	1	El escritor podrá ver de un solo vistazo todas las palabras que ha guardado en su diccionario personal.
	2	El escritor podrá añadir nuevas palabras que se añadirán a la lista de palabras.
Postcondición	El escritor observa que la nueva palabra añadida aparece en la lista de palabras de su diccionario personal.	
Excepciones	Paso	Acción
	1,2	Fallo en la sincronización con la base de datos que provoca que los datos no se carguen correctamente.
Importancia	Media	

Tabla 16 – Caso de uso CU06.1

- **RF06.2 | CU06.2 – El escritor podrá modificar palabras en su diccionario personal**

RF06.2	El escritor podrá modificar palabras en su diccionario personal.
Dependencias	<i>RF04, RF05, RF06.1</i>
Descripción	Los escritores tendrán acceso diccionario personal global donde podrán ir almacenando palabras que encuentren interesantes y útiles para posibles usos futuros en sus propias obras. Dichas palabras podrán ser modificadas por el propio escritor.
Datos específicos	- <i>El diccionario está compuesto por palabras y cada palabra podrá tener una descripción de su significado.</i>
Prioridad	Media
Comentarios	No tiene comentarios

Tabla 17 – Requisito funcional RF06.2

Código	CU06.2	
Requisito asociado	El escritor podrá modificar palabras en su diccionario personal.	
Objetivo asociado	El escritor podrá gestionar todas las palabras de su diccionario personal modificando las palabras existentes.	
Descripción	El escritor tendrá la posibilidad de gestionar su diccionario personal modificando el título o descripción de las palabras en su propio diccionario personal.	
Precondición	El escritor debe acceder al perfil para poder entrar a su propio diccionario personal.	
Secuencia normal	Paso	Acción
	1	El escritor podrá ver de un solo vistazo todas las palabras que ha guardado en su diccionario personal.
	2	El escritor podrá modificar cualquier palabra de su diccionario personal.
Postcondición	El escritor puede ver como la palabra aparece modificada.	
Excepciones	Paso	Acción
	1,2	Fallo en la sincronización con la base de datos que provoca que los datos no se carguen correctamente.
Importancia	Media	

Tabla 18 – Caso de uso CU06.2

- **RF06.3 | CU06.3 – El escritor podrá eliminar palabras en su diccionario personal**

RF06.3	El escritor podrá eliminar palabras en su diccionario personal.
Dependencias	<i>RF04, RF05, RF06.1, RF06.2</i>
Descripción	Los escritores tendrán acceso diccionario personal global donde podrán ir almacenando palabras que encuentren interesantes y útiles para posibles usos futuros en sus propias obras. Dichas palabras las podrá eliminar el propio escritor cuando crea oportuno.
Datos específicos	- <i>El diccionario está compuesto por palabras y cada palabra podrá tener una descripción de su significado.</i>
Prioridad	Media
Comentarios	No tiene comentarios

Tabla 19 – Requisito funcional RF06.3

Código	CU06.3	
Requisito asociado	El escritor podrá eliminar palabras en su diccionario personal.	
Objetivo asociado	El escritor podrá gestionar todas las palabras de su diccionario personal eliminando las ya existentes.	
Descripción	El escritor tendrá la posibilidad de gestionar su diccionario personal eliminando las palabras ya creadas de su diccionario personal.	
Precondición	El escritor debe acceder al perfil para poder entrar a su propio diccionario personal.	
Secuencia normal	Paso	Acción
	1	El escritor podrá ver de un solo vistazo todas las palabras que ha guardado en su diccionario personal.
	2	El escritor podrá eliminar las palabras que crea oportunas.
Postcondición	El escritor puede ver como la palabra seleccionada no aparece en la lista de palabras al haberse eliminado.	
Excepciones	Paso	Acción
	1,2	Fallo en la sincronización con la base de datos que provoca que los datos no se carguen correctamente.
Importancia	Media	

Tabla 20 – Caso de uso CU06.3

RF08 | CU08 - El escritor tendrá acceso a un glosario asociado a cada libro

RF08	El escritor tendrá acceso a un glosario asociado a cada libro
Dependencias	<i>RF01, RF02</i>
Descripción	El escritor tendrá acceso a un glosario de palabras asociado a cada libro que desee crear.
Datos específicos	<ul style="list-style-type: none"> ▪ <i>El glosario almacenará palabras propias que el autor decida usar en dicho libro y que serán útiles explicar para comprender en mayor profundidad la obra escrita.</i> ▪ <i>El formato del glosario debe ser como un diccionario, donde se guarde la palabra a explicar y una breve descripción.</i>
Prioridad	Media
Comentarios	No tiene comentarios

Tabla 21 – Requisito funcional RF08

Código	CU08	
Requisito asociado	El escritor tendrá acceso a un glosario asociado a cada libro	
Objetivo asociado	El escritor podrá tener un glosario por libro.	
Descripción	Cada libro tendrá un glosario por si el escritor tiene que definir alguna palabra propia que aparezca en el libro.	
Precondición	El escritor debe haber iniciado sesión en el sistema.	
Secuencia normal	Paso	Acción
	1	El escritor irá a su perfil si no lo está.
	2	Elegirá un libro al que quiera acceder del listado.
	2a	Si no tiene ningún libro creado, creará uno y lo seleccionará.
	4	Aparecen varias pestañas con información del libro, una de las pestañas será el glosario.
	5	Pinchar sobre la pestaña de glosario.
	6	Aparecerá el glosario de ese libro.
Postcondición	Pantalla del glosario del libro seleccionado.	
Excepciones	Paso	Acción
	2	No se selecciona el libro correctamente.
	5	No carga correctamente el glosario.
Importancia	Media	

Tabla 22 – Caso de uso CU08

RF09 - El escritor podrá crear, editar y borrar palabras en sus glosarios

- RF09.1 | CU09.1 - El escritor podrá crear palabras en sus glosarios

RF09.1	El escritor podrá crear palabras en sus glosarios
Dependencias	RF08
Descripción	El escritor podrá crear palabras en los glosarios asociados a sus libros.
Datos específicos	<ul style="list-style-type: none">▪ <i>El escritor podrá añadir palabras del glosario, el cual está compuesto por un título y su significado en el libro.</i>
Prioridad	Media
Comentarios	No tiene comentarios

Tabla 23 – Requisito funcional RF09.1

Código	CU09.1	
Requisito asociado	El escritor podrá crear palabras en sus glosarios	
Objetivo asociado	El escritor podrá añadir palabras en los glosarios.	
Descripción	Cada libro tendrá un glosario por si el escritor tiene que definir alguna palabra propia que aparezca en el libro. En estos glosarios, el escritor y autor del libro, podrá añadir nuevas palabras.	
Precondición	El escritor está en la pestaña del glosario al haber seleccionado un libro.	
Secuencia normal	Paso	Acción
	1	El escritor podrá ver a simple vista el listado de palabras del glosario asociado al libro seleccionado.
	2	El escritor podrá añadir palabras al glosario.
Postcondición	El escritor puede ver que la nueva palabra se ha añadido al glosario del libro seleccionado.	
Excepciones	Paso	Acción
	1,2	Algún problema de sincronización con la base de datos.
Importancia	Media	

Tabla 24 – Caso de uso CU09.1

- **RF09.2 | CU09.2 - El escritor podrá modificar palabras en sus glosarios**

RF09.2	El escritor podrá modificar palabras en sus glosarios
Dependencias	<i>RF09.1</i>
Descripción	El escritor podrá modificar palabras ya creadas de los glosarios asociados a sus libros.
Datos específicos	<ul style="list-style-type: none"> ▪ <i>El escritor podrá añadir palabras del glosario, el cual está compuesto por un título y su significado en el libro.</i>
Prioridad	<i>Media</i>
Comentarios	No tiene comentarios

Tabla 25 – Requisito funcional RF09.2.

Código	CU09.2	
Requisito asociado	El escritor podrá modificar palabras en sus glosarios	
Objetivo asociado	El escritor podrá modificar las palabras de sus glosarios.	
Descripción	Cada libro tendrá un glosario por si el escritor tiene que definir alguna palabra propia que aparezca en el libro. En estos glosarios, el escritor y autor del libro, podrá modificar palabras ya creadas.	
Precondición	El escritor está en la pestaña del glosario al haber seleccionado un libro.	
Secuencia normal	Paso	Acción
	1	El escritor podrá ver a simple vista el listado de palabras del glosario asociado al libro seleccionado.
	2	El escritor podrá modificar las palabras ya creadas.
Postcondición	El escritor puede ver que se ha modificado la palabra seleccionada del glosario.	
Excepciones	Paso	Acción
	1,2	Algún problema de sincronización con la base de datos.
Importancia	Media	

Tabla 26 – Caso de uso CU09.2

- **RF09.3 | CU09.3 - El escritor podrá eliminar palabras en sus glosarios**

RF09.3	El escritor podrá eliminar palabras en sus glosarios
Dependencias	<i>RF09.1, RF09.2</i>
Descripción	El escritor podrá eliminar palabras de los glosarios asociados a sus libros.
Datos específicos	<ul style="list-style-type: none"> ▪ <i>El escritor podrá añadir palabras del glosario, el cual está compuesto por un título y su significado en el libro.</i>
Prioridad	Media
Comentarios	No tiene comentarios

Tabla 27 – Requisito funcional RF09.3

Código	CU09.3	
Requisito asociado	El escritor podrá eliminar palabras en sus glosarios	
Objetivo asociado	El escritor podrá eliminar cualquier palabra creada en alguno de sus glosarios.	
Descripción	Cada libro tendrá un glosario por si el escritor tiene que definir alguna palabra propia que aparezca en el libro. En estos glosarios, el escritor y autor del libro, podrá eliminar palabras ya creadas.	
Precondición	El escritor está en la pestaña del glosario al haber seleccionado un libro.	
Secuencia normal	Paso	Acción
	1	El escritor podrá ver a simple vista el listado de palabras del glosario asociado al libro seleccionado.
	2	El escritor podrá eliminar las palabras ya creadas.
Postcondición	El escritor podrá ver como la palabra seleccionada ya no aparece en el glosario.	
Excepciones	Paso	Acción
	1,2	Algún problema de sincronización con la base de datos.
Importancia	Media	

Tabla 28 – Caso de uso CU09.3

RF10 | CU10 - El escritor tendrá acceso a notas asociadas a cada libro

RF10	El escritor tendrá acceso a notas asociadas a cada libro
Dependencias	<i>RF01, RF02</i>
Descripción	El escritor tendrá acceso a unas notas asociadas a cada libro que desee crear.
Datos específicos	<ul style="list-style-type: none"> ▪ <i>Cada libro tendrá una lista de notas que el escritor desee añadir para aclarar cualquier cosa del libro.</i>
Prioridad	Media
Comentarios	No tiene comentarios

Tabla 29 – Requisito funcional RF10

Código	CU10	
Requisito asociado	El escritor tendrá acceso a unas notas asociado a cada libro	
Objetivo asociado	El escritor podrá tener unas notas por libro.	
Descripción	Cada libro tendrá unas notas por si el escritor tiene que aclarar algo que aparezca en el libro.	
Precondición	El escritor debe haber iniciado sesión en el sistema.	
Secuencia normal	Paso	Acción
	1	El escritor irá a su perfil si no lo está.
	2	Elegirá un libro al que quiera acceder del listado.
	2a	Si no tiene ningún libro creado, creará uno y lo seleccionará.
	4	Aparecen varias pestañas con información del libro, una de las pestañas será la de notas.
	5	Pinchar sobre la pestaña de notas.
	6	Aparecerá las notas de ese libro.
Postcondición	Pantalla de notas del libro seleccionado.	
Excepciones	Paso	Acción
	2	No se selecciona el libro correctamente.
	5	No carga correctamente el glosario.
Importancia	Media	

Tabla 30 – Caso de uso CU10

RF11 - El escritor podrá crear, editar y borrar notas en su lista de notas

- RF11.1 | CU11.1 - El escritor podrá crear notas en su lista de notas

RF11.1	El escritor podrá crear notas en su lista de notas
Dependencias	<i>RF10</i>
Descripción	El escritor podrá crear notas en su lista de notas asociadas a cada libro.
Datos específicos	- <i>Las notas estarán compuestas por un título identificativo y una descripción.</i>
Prioridad	<i>Media</i>
Comentarios	No tiene comentarios

Tabla 31 – Requisito funcional RF11.1

Código	CU11.1	
Requisito asociado	El escritor podrá crear notas en su lista de notas.	
Objetivo asociado	El escritor podrá añadir notas a las listas de notas que cada libro tiene asociado.	
Descripción	El escritor podrá gestionar sus listas de notas asociadas a cada libro pudiendo añadir notas explicativas.	
Precondición	El escritor está en la pestaña de notas al haber seleccionado un libro.	
Secuencia normal	Paso	Acción
	1	El escritor ver todas las notas relacionadas con el libro seleccionado.
	2	El escritor podrá añadir nuevas notas al libro seleccionado.
Postcondición	El escritor puede ver la nueva nota añadida.	
Excepciones	Paso	Acción
	1,2	Algún problema de sincronización con la base de datos.
Importancia	Media	

Tabla 32 – Caso de uso CU11.1

- **RF11.2 | CU11.2 - El escritor podrá modificar notas de su lista de notas**

RF11.2	El escritor podrá modificar notas en su lista de notas
Dependencias	<i>RF10, RF11.1</i>
Descripción	El escritor podrá modificar las notas de su lista de notas asociadas a cada libro.
Datos específicos	- <i>Las notas estarán compuestas por un título identificativo y una descripción.</i>
Prioridad	<i>Media</i>
Comentarios	No tiene comentarios

Tabla 33 – Requisito funcional RF11.2

Código	CU11.2	
Requisito asociado	El escritor podrá modificar notas de su lista de notas.	
Objetivo asociado	El escritor podrá modificar las notas de sus libros.	
Descripción	El escritor podrá gestionar sus listas de notas asociadas a cada libro pudiendo modificar dichas notas para cambiar la explicación o título.	
Precondición	El escritor está en la pestaña de notas al haber seleccionado un libro.	
Secuencia normal	Paso	Acción
	1	El escritor ver todas las notas relacionadas con el libro seleccionado.
	1a	El escritor podrá modificar cualquier palabra de la lista de notas asociada al libro seleccionado.
Postcondición	El escritor puede ver la nota modificada en lista de notas.	
Excepciones	Paso	Acción
	1, 2	Algún problema de sincronización con la base de datos.
Importancia	Media	

Tabla 34 – Caso de uso CU11.2

- **RF11.3 | CU11.3 - El escritor podrá eliminar notas de su lista de notas**

RF11.3	El escritor podrá eliminar notas de su lista de notas
Dependencias	<i>RF10, RF11.1, RF11.2</i>
Descripción	El escritor podrá eliminar notas de su lista de notas asociadas a cada libro.
Datos específicos	- <i>Las notas estarán compuestas por un título identificativo y una descripción.</i>
Prioridad	Media
Comentarios	No tiene comentarios

Tabla 35 – Requisito funcional RF11.3

Código	CU11.3	
Requisito asociado	El escritor podrá eliminar notas de su lista de notas.	
Objetivo asociado	El escritor podrá eliminar notas de sus listas de notas.	
Descripción	El escritor podrá gestionar sus listas de notas asociadas a cada libro pudiendo eliminar dichas notas que ya no sean de utilidad.	
Precondición	El escritor está en la pestaña de notas al haber seleccionado un libro.	
Secuencia normal	Paso	Acción
	1	El escritor ver todas las notas relacionadas con el libro seleccionado.
	1a	El escritor podrá eliminar las notas de la lista de notas asociadas al libro elegido.
Postcondición	El escritor puede ver como se ha eliminado la nota de la lista.	
Excepciones	Paso	Acción
	1,2	Algún problema de sincronización con la base de datos.
Importancia	Media	

Tabla 35 – Caso de uso CU11.3

RF12 | CU12 - El escritor tendrá acceso a hojas de personajes asociadas a sus libros

RF12	El escritor tendrá acceso a hojas de personajes asociadas a sus libros
Dependencias	<i>Sin dependencias</i>
Descripción	El escritor podrá acceder a una galería donde encontrará todos los personajes creados.
Datos específicos	- <i>Podrá filtrar los personajes por libro.</i>
Prioridad	Media
Comentarios	No tiene comentarios

Tabla 36 – Requisito funcional RF12

Código	CU12	
Requisito asociado	El escritor tendrá acceso a hojas de personajes asociadas a sus libros	
Objetivo asociado	Galería con todos los personajes creados.	
Descripción	El escritor podrá ver todos los personajes creados que componen sus obras, filtrando por libro al que pertenecen.	
Precondición	El escritor accederá a la galería pinchando en la barra de navegación en el enlace “Diseño de personajes”	
Secuencia normal	Paso	Acción
	1	El escritor podrá ver todos los personajes sin ningún filtro.
	1a	El escritor no verá personajes al no tener ninguno creado.
	2	El escritor podrá filtrar los personajes por libro.
	2a	El escritor verá los personajes filtrados por el libro seleccionado.
	2b	El escritor no verá ningún personaje ya que el libro seleccionado no tiene aún personajes.
Postcondición	El escritor tiene acceso a la galería de personajes.	
Excepciones	Paso	Acción
	1, 1a, 2, 2a, 2b	Algún problema de sincronización con la base de datos.
Importancia	Media	

Tabla 37 – Caso de uso CU12

RF13 - El escritor podrá crear, editar y borrar sus hojas de personajes

- RF13.1 | CU13.1- El escritor podrá crear hojas de personajes

RF13.1	El escritor podrá crear hojas de personajes
Dependencias	RF12
Descripción	El escritor podrá crear hojas de personajes.
Datos específicos	- <i>Los personajes de la galería serán seleccionables para ver su información.</i>
Prioridad	Media
Comentarios	No tiene comentarios

Tabla 38 – Requisito funcional RF13.1

Código	CU13.1	
Requisito asociado	El escritor podrá crear hojas de personajes	
Objetivo asociado	El escritor podrá añadir personajes a su galería de personajes.	
Descripción	El escritor podrá gestionar su galería de personajes añadiendo los personajes necesarios.	
Precondición	El escritor accederá a la galería de personajes a través del diseño de personajes.	
Secuencia normal	Paso	Acción
	1	El escritor podrá ver todos los personajes creados a simple vista.
	2	El escritor podrá añadir nuevos personajes asociados a sus libros creados
Postcondición	El escritor podrá ver el nuevo personaje creado en la galería de personajes.	
Excepciones	Paso	Acción
	1,2	Algún problema de sincronización con la base de datos o carga de la página.
Importancia	Media	

Tabla 39 – Caso de uso CU13.1

- **RF13.2 | CU13.2 - El escritor podrá modificar las hojas de personajes**

RF13.2	El escritor podrá modificar las hojas de personajes
Dependencias	<i>RF12, RF13.1</i>
Descripción	El escritor podrá modificar las hojas de personajes.
Datos específicos	- <i>Los personajes de la galería serán seleccionables para ver su información.</i>
Prioridad	Media
Comentarios	No tiene comentarios

Tabla 39 – Requisito funcional RF13.2

Código	CU13.2	
Requisito asociado	El escritor podrá modificar las hojas de personajes	
Objetivo asociado	El escritor podrá modificar los personajes de su galería de personajes.	
Descripción	El escritor podrá gestionar su galería de personajes modificando los personajes ya creados.	
Precondición	El escritor accederá a la galería de personajes a través del diseño de personajes.	
Secuencia normal	Paso	Acción
	1	El escritor podrá ver todos los personajes creados a simple vista.
	2	El escritor podrá modificar los personajes de su galería de personajes.
Postcondición	El escritor podrá ver el personaje modificado en la galería de personajes.	
Excepciones	Paso	Acción
	1,2	Algún problema de sincronización con la base de datos o carga de la página.
Importancia	Media	

Tabla 40 – Caso de uso CU13.2

- **RF13.3 | CU13.3 - El escritor podrá eliminar las hojas de personajes**

RF13.3	El escritor podrá eliminar las hojas de personajes
Dependencias	<i>RF12, RF13.1, RF13.2</i>
Descripción	El escritor podrá eliminar las hojas de personajes.
Datos específicos	<ul style="list-style-type: none"> - <i>Los personajes de la galería serán seleccionables para ver su información.</i> - <i>Se pedirá confirmación de borrado.</i>
Prioridad	Media
Comentarios	No tiene comentarios

Tabla 41 – Requisito funcional RF13.3

Código	CU13.3	
Requisito asociado	El escritor podrá eliminar las hojas de personajes	
Objetivo asociado	El escritor podrá eliminar los personajes de su galería de personajes.	
Descripción	El escritor podrá gestionar su galería de personajes eliminando los personajes que no necesite.	
Precondición	El escritor accederá a la galería de personajes a través del diseño de personajes.	
Secuencia normal	Paso	Acción
	1	El escritor podrá ver todos los personajes creados a simple vista.
	2	El escritor podrá eliminar los personajes de su galería de personajes.
Postcondición	El escritor podrá ver el personaje eliminado de la galería de personajes.	
Excepciones	Paso	Acción
	1,2	Algún problema de sincronización con la base de datos o carga de la página.
Importancia	Media	

Tabla 42 – Caso de uso CU13.3

RF24 | CU24 - El escritor podrá acceder a una lista de tareas para planificarse

RF14	El escritor podrá acceder a una lista de tareas para planificarse
Dependencias	<i>No tiene dependencias</i>
Descripción	El escritor acceder a una sección donde pueda ver todas las tareas creadas para poder organizarse y llevar a cabo una planificación para optimizar tiempos a la hora de crear sus obras.
Datos específicos	<i>Sin datos específicos</i>
Prioridad	Media
Comentarios	No tiene comentarios

Tabla 43 – Requisito funcional RF24

Código	CU24	
Requisito asociado	El escritor podrá acceder a una lista de tareas para planificarse	
Objetivo asociado	El escritor podrá organizarse y planificar todo el ciclo de escritura de sus obras.	
Descripción	El escritor podrá acceder a una sección donde podrá ver todas las tareas creadas, el estado en el que están, fecha de fin, entre otras características.	
Precondición	El escritor accederá a la sección de tareas pinchando en la barra de navegación en el enlace “Planificación”.	
Secuencia normal	Paso	Acción
	1	El escritor puede ver todas las tareas que ha creado a simple vista.
	2	El escritor puede ver las tareas que tiene creadas y tienen el estado de pendiente.
	3	El escritor puede ver las tareas que tiene creadas y tienen el estado de en curso, son las tareas que están realizando.
	2a	El escritor puede ver las tareas que tiene creadas y tienen el estado de completada, son las tareas que ya ha cerrado el escritor.
	3	El escritor podrá borrar cualquier personaje que haya creado, se le pedirá confirmación de borrad.
Postcondición	El escritor podrá organizarse las tareas para una correcta planificación a la hora de escribir sus obras.	
Excepciones	Paso	Acción
	1, 1a ,2, 2a,2b	Algún problema de sincronización con la base de datos o carga de la página.
Importancia	Media	

Tabla 43 – Caso de uso CU24

RF25 - El escritor podrá crear, editar y borrar tareas en su lista de tareas

- RF25.1 | CU25.1 - El escritor podrá crear tareas en su lista de tareas

RF25.1	El escritor podrá crear tareas en su lista de tareas
Dependencias	<i>RF24</i>
Descripción	El escritor acceder a una sección donde pueda ver todas las tareas creadas para poder organizarse creando tareas a realizar.
Datos específicos	<i>Sin datos específicos</i>
Prioridad	Media
Comentarios	No tiene comentarios

Tabla 44 – Requisito funcional RF25.1

Código	CU25.1	
Requisito asociado	El escritor podrá crear tareas en su lista de tareas.	
Objetivo asociado	El escritor podrá añadir tareas a su lista de tareas.	
Descripción	El escritor podrá organizarse añadiendo tareas a su lista de tareas.	
Precondición	El escritor accederá a la sección de tareas a través de la sección de planificación.	
Secuencia normal	Paso	Acción
	1	El escritor puede ver todas las tareas que ha creado a simple vista.
	2	El escritor podrá crear nuevas tareas
Postcondición	El escritor podrá ver la nueva tarea añadida a la lista de tareas.	
Excepciones	Paso	Acción
	1,2	Algún problema de sincronización con la base de datos o carga de la página.
Importancia	Media	

Tabla 45 – Caso de uso CU25.1

- **RF25.2 | CU25.2 - El escritor podrá modificar las tareas de su lista de tareas**

RF25.2	El escritor podrá modificar las tareas de su lista de tareas
Dependencias	<i>RF24, RF25.1</i>
Descripción	El escritor acceder a una sección donde pueda ver todas las tareas creadas para poder organizarse modificando las tareas ya creadas.
Datos específicos	<i>Sin datos específicos</i>
Prioridad	Media
Comentarios	No tiene comentarios

Tabla 46 – Requisito funcional RF25.2

Código	CU25.2	
Requisito asociado	El escritor podrá modificar las tareas de su lista de tareas.	
Objetivo asociado	El escritor podrá modificar sus tareas.	
Descripción	El escritor podrá organizarse modificando las tareas de su lista de tareas.	
Precondición	El escritor accederá a la sección de tareas a través de la sección de planificación.	
Secuencia normal	Paso	Acción
	1	El escritor puede ver todas las tareas que ha creado a simple vista.
	2	El escritor podrá modificar las tareas que tiene en su lista de tareas.
Postcondición	El escritor podrá ver las modificaciones de la tarea en lista de tareas.	
Excepciones	Paso	Acción
	1,2	Algún problema de sincronización con la base de datos o carga de la página.
Importancia	Media	

Tabla 47 – Caso de uso CU25.2

- **RF25.3 | CU25.3 - El escritor podrá eliminar las tareas de su lista de tareas**

RF25.3	El escritor podrá eliminar las tareas de su lista de tareas
Dependencias	<i>RF24, RF25.1, RF25.2</i>
Descripción	El escritor acceder a una sección donde pueda ver todas las tareas creadas para poder organizarse eliminando tareas ya creadas.
Datos específicos	<i>Sin datos específicos</i>
Prioridad	Media
Comentarios	No tiene comentarios

Tabla 48 – Requisito funcional RF25.3

Código	CU25.3	
Requisito asociado	El escritor podrá eliminar las tareas de su lista de tareas.	
Objetivo asociado	El escritor podrá eliminar tareas.	
Descripción	El escritor podrá organizarse eliminando las tareas de su lista de tareas.	
Precondición	El escritor accederá a la sección de tareas a través de la sección de planificación.	
Secuencia normal	Paso	Acción
	1	El escritor puede ver todas las tareas que ha creado a simple vista.
	2	El escritor podrá eliminar tareas que tiene en su lista de tareas.
Postcondición	El escritor podrá ver que tarea eliminada no aparece en lista de tareas.	
Excepciones	Paso	Acción
	1,2	Algún problema de sincronización con la base de datos o carga de la página.
Importancia	Media	

Tabla 49 – Caso de uso CU25.3

RF27 | CU27 - El escritor dispondrá de análisis de sentimientos de sus obras escritas.

RF27	El escritor dispondrá de análisis de sentimientos de sus obras escritas.
Dependencias	<i>Sin dependencias.</i>
Descripción	El escritor acceder a una sección donde pueda elegir cualquiera de los libros s creados por él mismo y realizarle un análisis de sentimientos.
Datos específicos	<i>El escritor recibirá una retroalimentación sobre sus obras que decida analizar, obteniendo si transmite una sensación positiva, negativa o no transmite ningún sentimiento (neutro).</i>
Prioridad	Media
Comentarios	No tiene comentarios

Tabla 50 – Requisito funcional RF27

Código	CU27	
Requisito asociado	El escritor dispondrá de análisis de sentimientos de sus obras escritas.	
Objetivo asociado	El escritor podrá obtener una retroalimentación sobre los sentimientos que transmiten las obras.	
Descripción	El escritor al acceder a la sección de sentimientos podrá ver y analizar las obras creadas por él.	
Precondición	El escritor accederá a la sección de análisis de sentimientos.	
Secuencia normal	Paso	Acción
	1	El escritor podrá elegir el libro que desea analizar.
	1a	Si el libro seleccionado tiene algún análisis realizado con anterioridad se le mostrará al escritor.
	2	El escritor podrá ejecutar los análisis.
	2a	El escritor recibirá una retroalimentación del análisis realizado al libro seleccionado.
Postcondición	El escritor podrá observar y analizar cuáles son los sentimientos que está transmitiendo con la obra.	
Excepciones	Paso	Acción
	1, 1a,2, 2a	Algún problema de sincronización con la base de datos o carga de la página.
Importancia	Media	

Tabla 51 – Caso de uso CU27

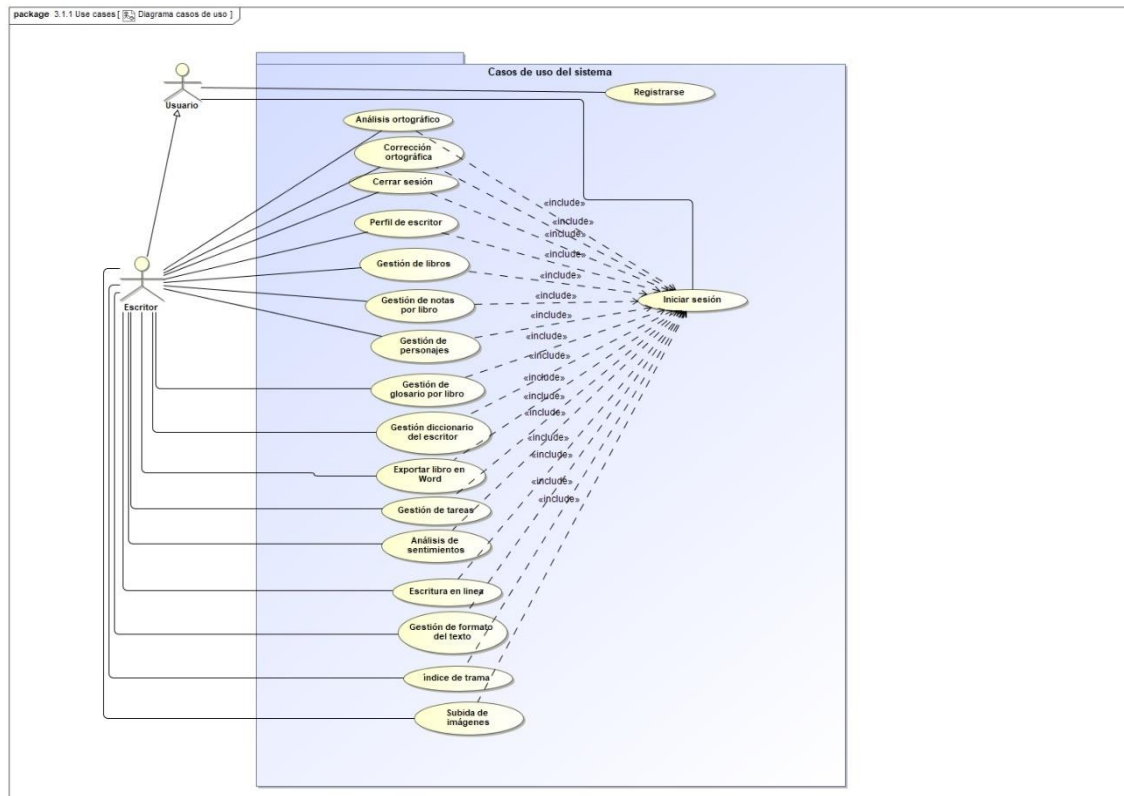


Figura 1 – Casos de uso del sistema

Para cerrar la explicación de los requisitos y casos de uso de forma individual, se han representado todos los casos de uso del sistema para poder entenderlo y observarlo de una forma más amigable. En este caso se ha utilizado MagicDraw como programa para hacer los diagramas a lo largo de este proyecto.

Como se puede ver en el diagrama los actores principales, como ya se ha mencionado anteriormente, son el usuario no registrado y el escritor, usuario registrado.

Todas las funcionalidades requieren del usuario registrado, el escritor, inicie sesión con las credenciales correctas con las que se registró en el sistema, así podrá utilizarlas sin ningún problema.

Todos los objetivos que se querían abordar desde un primer momento y que son los tratados en este proyecto se han definido en MagicDraw, al igual que todos los requisitos funcionales derivados de dichos objetivos y, además todos los casos de uso. Por lo tanto, se han definido unas matrices de dependencias cruzando objetivos y requisitos funcionales, requisitos funcionales y casos de uso y finalmente objetivos con casos de uso. Se han cruzado y marcado obteniendo la trazabilidad completa desde el objetivo hasta el caso de uso que lo resuelve, pasando por el requisito funcional.

3.1 Functional requirements		FR-01	FR-02	FR-03	FR-04	FR-05	FR-06	FR-06.1	FR-06.2	FR-06.3	FR-07	FR-08	FR-09	FR-10	FR-10.1	FR-10.2	FR-10.3	FR-10	FR-11	FR-11.1	FR-11.2	FR-11.3	FR-12	FR-13	FR-14	FR-14.1	FR-14.2	FR-14.3	FR-14	FR-15.1	FR-15.2	FR-15.3	FR-15	FR-16	FR-17.1	FR-17.2	FR-17.3	FR-18	FR-19	FR-20	FR-21	FR-22	FR-23	FR-24	FR-25	FR-25.1	FR-25.2	FR-25.3	FR-26	FR-27																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
FR-01 Registro de usuarios		1																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
FR-02 Inicio de sesión		1	1																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															
FR-03 Cierre de sesión		1	1	1																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																														
FR-04 Perfil del escritor		1	1	1	1																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
FR-05 Acceso diccionario persona		1	1	1	1	1																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												

Figura 2 – Matriz de dependencia Objetivos – Requisitos funcionales

[illegible]

Casos de uso del sistema [3.1 Función]		3.1 Functional requirements	
○ Acceso a las notas por libro	5	FR-01 Registro de usuarios	1
○ Análisis de sentimientos	1	FR-02 Inicio de sesión	1
○ Análisis ortográfico	2	FR-03 Cierre de sesión	1
○ Cerrar sesión	1	FR-04 Perfil del escritor	1
○ Corrección ortográfica	2	FR-05 Acceso diccionario personal	1
○ Escritura en línea	7	FR-06 CRUD Diccionario personal	1
○ Exportar libro en Word	1	FR-06.1 Creación de palabras	1
○ Gestión de formato del texto	1	FR-06.2 Actualización de palabras	1
○ Gestión de glosario por libro	5	FR-06.3 Borrado de palabras	1
○ Gestión de libros	6	FR-07 Visualizar información por libro	1
○ Gestión de notas por libro	5	FR-08 Glosario por libro	1
○ Gestión de personajes	5	FR-09 CRUD Glosario	1
○ Gestión de tareas	5	FR-10.1 Creación palabras	1
○ Gestión diccionario del escritor	5	FR-10.2 Actualización de palabras	1
○ Índice de trama	1	FR-10.3 Borrado de palabras	1
○ Iniciar sesión	1	FR-10 Acceso a notas por libro	2
○ Perfil de escritor	1	FR-11 CRUD Notas	2
○ Registrarse	1	FR-11.1 Creación de notas	2
○ Subida de imágenes	1	FR-11.2 Actualización de notas	2
		FR-11.3 Borrado de notas	2
		FR-12 Acceso al diseño de hojas de p	1
		FR-13 CRUD Hoja de personajes	1
		FR-14.1 Creación de hoja de pers	1
		FR-14.2 Actualización de hoja de	1
		FR-14.3 Borrado de hoja de pers	1
		FR-14 CRUD Libros	1
		FR-15.1 Creación de libros	1
		FR-15.2 Actualización de libros	1
		FR-15.3 Borrado de libros	1
		FR-15 Visualización de capítulos	1
		FR-16 CRUD Capítulos	1
		FR-17.1 Creación de capítulos	1
		FR-17.2 Actualización de capítulos	1
		FR-17.3 Borrado de capítulos	1
		FR-17 Acceso escritura en línea	1
		FR-18 Escritura de capítulos en línea	1
		FR-19 Elecciones de formato texto	2
		FR-20 Exportar los capítulos en format	1
		FR-21 Corrección ortográfica	2
		FR-22 Análisis ortográfico	2
		FR-23 Índice de acontecimientos escr	2
		FR-24 Acceso a lista de tareas	1
		FR-25 CRUD Lista de tareas	1
		FR-25.1 Creación de tareas	1
		FR-25.2 Actualización de tareas	1
		FR-25.3 Borrado de tareas	1
		FR-26 Subida de imágenes	1
		FR-27 Análisis de sentimientos	1

3

Metodología

Para la realización de esta plataforma, que se trata de un trabajo en grupo, se ha tomado la decisión de utilizar la plataforma de Microsoft, como lo es Azure DevOps¹. Esta plataforma además de proporcionar soporte como repositorio de código con Git², ofrece otras muchas herramientas, pero se centrará en la herramienta que ofrece para la planificación de proyectos. Entre esta herramienta se encuentra un Kanban para la gestión y planificación de proyectos ofreciendo varios paneles en los que revisar y planificar las tareas, historias de usuario, iteraciones, etc.

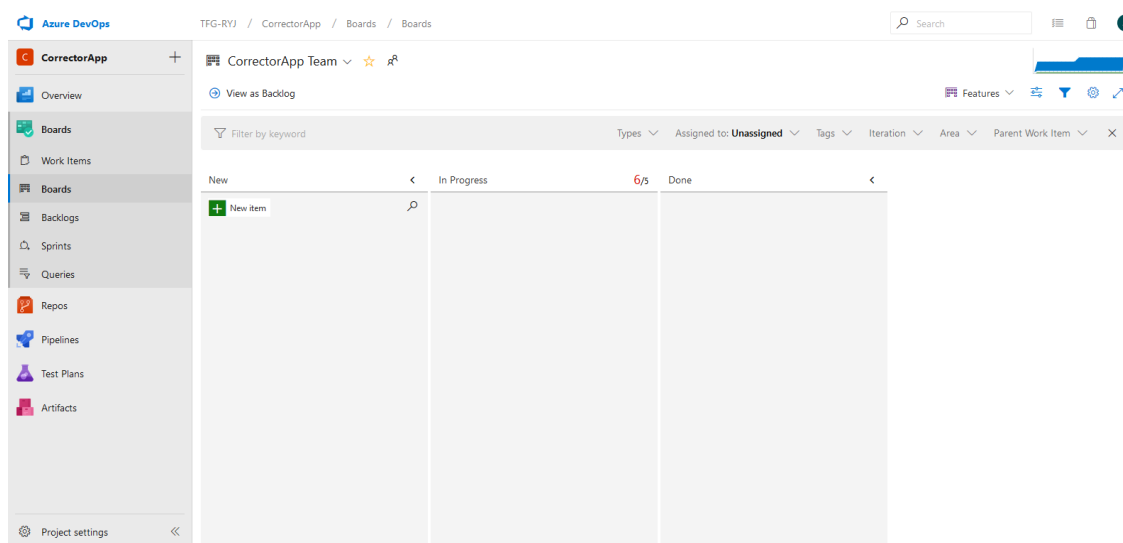


Figura 5 – Azure DevOps Kanban

Planteamiento

Para la planificación de este proyecto se ha decidido usar una metodología ágil como es Scrum, ya que esta herramienta incita a usar dicha metodología y se adapta de forma sencilla e integrándose de forma muy cómoda para trabajar en equipo.

Entre otras de las herramientas para la gestión de proyectos candidatas fueron Jira o Trello, pero por cercanía a su uso, previos conocimientos y por qué Azure DevOps se utiliza como repositorio de código, salió vencedora esta última.

Al utilizar una metodología ágil como es Scrum³ se decidió delimitar el tiempo de las iteraciones, más conocidas como sprints, a dos semanas. Donde en esas dos semanas cada integrante del equipo debería de ver las tareas de la lista de

tareas pendientes que aún no se han realizado y estimar su desarrollo para intentar que sea lo más exacto posible. Las tareas estimadas y elegidas por cada integrante deben de completarse en el mismo sprint para no repercutir en la planificación global. A medida que se va avanzando en el desarrollo podrían darse errores en desarrollos anteriores, más conocidos como *bugs*, entonces debería abrirse un *bug*, o incidencia, a la tarea correspondiente para que dicha incidencia sea resuelta lo antes posible.

Al no haber Product Owner los dos integrantes del equipo se encargarán de dar prioridad a las tareas, priorizando las que se crean más oportunas y vitales para el correcto progreso del proyecto.

También son conocidas las reuniones diarias entre los integrantes del equipo para poner en común en qué tarea se encuentra cada desarrollador, tanto por ciento del desarrollo que lleva completado, si llegará de forma óptima a la estimación dada o hay que reestimar dicha tarea y así tener una foto global de la iteración en la que se encuentra el equipo.

Experiencia

Para la planificación, estas reuniones han sido constantes entre los dos integrantes del equipo, cada vez que uno empieza un desarrollo nuevo, se encuentra con problemas con la tarea activa o tiene alguna noticia que comentar con el otro integrante se realizan reuniones muy frecuentes.

En referencia a los *sprints* o iteraciones, ha sido complicado llevarlo a cabo al pie de la letra por incompatibilidad horaria por jornada laboral de los dos integrantes. Pero cada requisito al planificarse en tareas pequeñas se podría haber ajustado a la planificación de dos semanas de duración en los sprints. También se han encontrado con tareas infravaloradas que afectan a la planificación, ya que hubiesen ocupado dos sprints al reevaluar dicha tarea. Se han creado los requisitos en dicha plataforma de Azure DevOps y, para cada requisito se asignaron las historias de usuario adecuadas más pequeñas, para que, a su vez, se simplifiquen dichas historias de usuario en tareas que puedan cumplirse en una iteración de dos semanas.

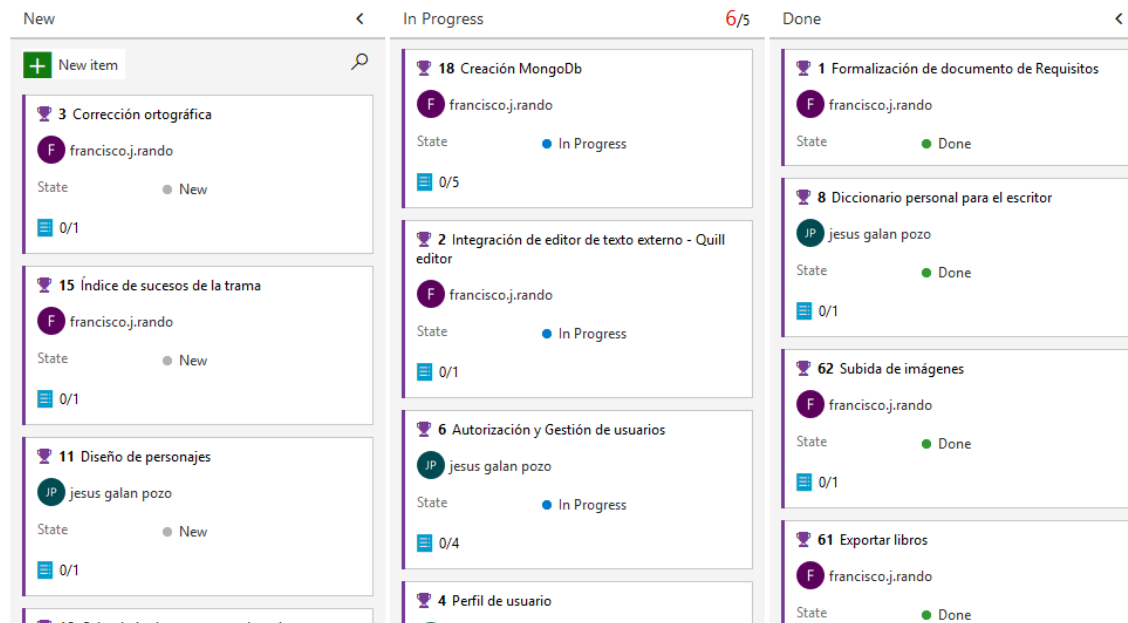


Figura 6 – Requisitos en Azure DevOps I

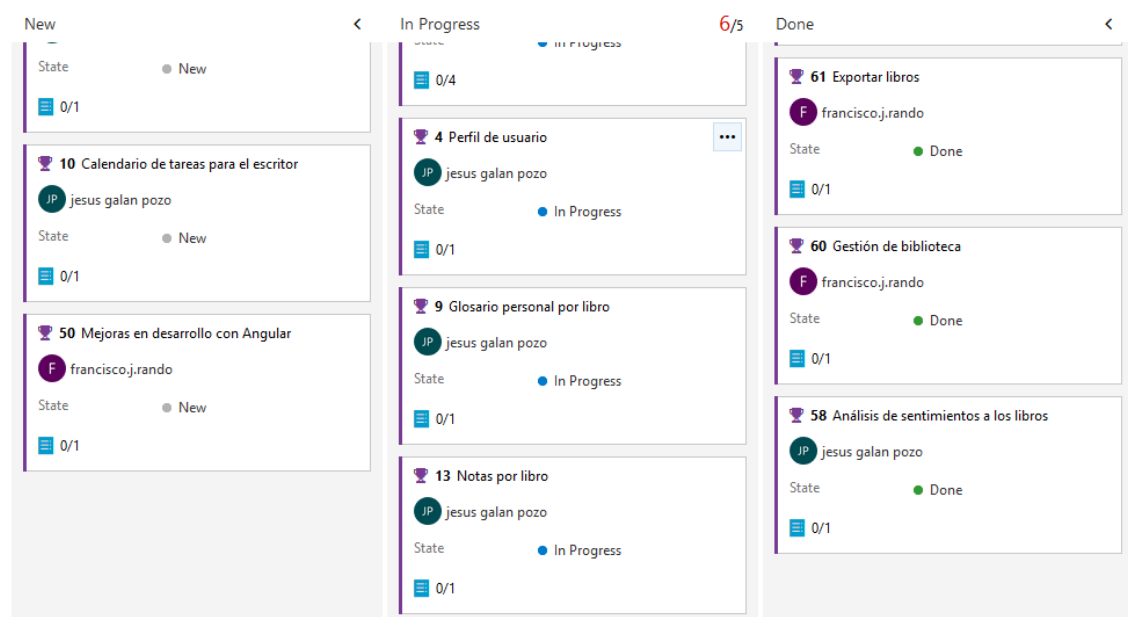


Figura 7 – Requisitos en Azure DevOps II

4

Elección de tecnologías

En este apartado se va a tratar todo lo relacionado con las tecnologías y plataformas que se van a utilizar para el desarrollo de este proyecto. En cada tecnología se explicará más en detalle apartados como una definición de la propia tecnología o plataforma, la motivación por la que usar dicha tecnología y no cualquier otra y las aportaciones de esta tecnología al proyecto y por último la experiencia a lo largo del desarrollo del proyecto.

Azure DevOps

Definición

Esta plataforma pertenece a Microsoft que ofrece una multitud de herramientas las cuales cubren todas las fases del desarrollo del software, desde la planificación de un proyecto hasta la supervisión y el control del producto una vez la aplicación ya está activa y dispuesta para el uso.

Motivación

Se ha seleccionado esta plataforma por todas la facilidades y herramientas que ofrece para llevar todo el control en cada una de las fases del desarrollo del software. Conocer con anterioridad esta plataforma y como trabajar con ella ha sido un aliciente para su elección. Para ser más concretos este proyecto va a utilizar Azure DevOps para las dos primeras etapas, planificación y desarrollo. Para estas dos etapas se hará uso del planificador de tareas a través del Kanban que nos proporciona la plataforma y en la parte de desarrollo se utilizará la herramienta de control de versiones que trae incorporada con Git.

Experiencia

Estas herramientas son de gran utilidad y valor al estar en una plataforma en la nube ya que proporciona una comodidad y facilidad de uso para llevar a cabo trabajos en grupo como lo es este proyecto. Ayudan a llevar una planificación más eficaz ya que todo el equipo puede ver, en el caso de la planificación, todas las tareas programas pendientes, las que se están realizando y las ya terminadas o están siendo probadas. Además, el control de versiones Git es una herramienta imprescindible hoy en día en cualquier proyecto de cierta envergadura, aportando gran valor a la hora de revisiones de código y todos los imprevistos que puedan ocurrir en el tiempo en el que se está desarrollando la aplicación.

Azure CosmosDB⁴

Definición

Es un servicio que está también en manos de Microsoft y sus servicios alojados en la nube, en concreto este servicio trata de una base de datos no relacional multimodelo la cual permite desacoplar la administración de la base de datos en máquinas remotas en la nube.

Motivación

Este servicio permite tener la base de datos de la plataforma de forma remota en la nube, por lo tanto, quita la desventaja de tener bases de datos replicadas en las máquinas locales de los desarrolladores. Teniendo este servicio de bases de datos centralizada en la nube permite tener una escalabilidad y flexibilidad que de otra forma sería mucho más difícil y tedioso. Azure ofrece una multitud de servicios de base de datos y la elección de CosmosDB ha sido por ser una base de datos no relacional y permitir el almacenamiento de documentos. De todos los modelos que se pueden elegir en CosmosDB se ha decidido usar MongoDB, que tiene unas características de lectura/escrituras muy buenas en tono a diez milisegundos con una altísima disponibilidad.

Experiencia

La elección de este servicio de la plataforma de Azure ha permitido un desarrollo más ágil y confortable al permitir tener una única base de datos centralizada en la que los desarrolladores pueden acceder de forma remota desde cualquier lugar y dispositivo. Además, la plataforma de Azure ofrece una interfaz de administración clara y sencilla donde no hace falta acceder a ninguna terminal. Al principio se encontraron algunas dificultades a la hora de la gestión de la colección al ser una base de datos no relacional. Los principales conflictos se obtuvieron en la gestión que hace la propia base de datos de los identificadores únicos y por lo tanto se tomó la decisión de tratar unos identificadores internos de manera manual. Para la integración con la parte back-end de .Net Core⁵ se encontraron problemas a la hora de mapear los documentos almacenados en la base de datos como Json⁶, por lo tanto, en los modelos se indicó de forma explícita a través de atributos la propiedad exacta como debía ser leída para una correcta comunicación con la base de datos y el back-end.

Swagger⁷

Definición

Swagger es una serie de reglas, especificaciones y herramientas ayudan a documentar las Apis. De esta manera, se puede realizar documentación que sea realmente útil. Este marco de trabajo o comúnmente llamado framework, ayuda a crear documentación que todo el mundo entienda.

Motivación

Este marco de trabajo ofrece una serie de características para documentar la Api que resulta muy útil, llamativo y muy cómodo a la hora de interactuar y probar con dicha Api. Además de poder generar una buena documentación de la Api ofrece una herramienta como SwaggerUI, una interfaz muy acogedora y sencilla donde el desarrollador puede ver en todo momento los endpoints o interfaces a las que poder atacar la Api y sus operaciones disponibles.

Experiencia

La integración de este framework ha sido relativamente sencilla ya que .Net Core dispone de su propio paquete oficial solo es necesario instalarlo en la solución y realizar una serie de configuraciones para que el framework consiga mapear los modelos de datos y las interfaces con sus operaciones.

ASP.Net Core Web Api

Definición

.Net Core es un nuevo marco de trabajo desarrollado por Microsoft el cual es multiplataforma y de alto rendimiento. Es una versión distinta al ASP.Net Framework clásico muy popular en su ámbito.

Motivación

Este marco de trabajo es una nueva versión mejorada ya que es multiplataforma, es decir, puede desarrollarse en Windows, MacOS y Linux y es de código abierto. La arquitectura en la que está compuesto es de forma modular, ofrece un núcleo básico para poder desarrollar aplicaciones, pero se puede ir completando este núcleo añadiendo extensiones o librerías externas, más conocidas como paquete Nugets⁸. Es un caso unificado para crear Api web y una interfaz web de usuarios. Ofrece al desarrollador un marco de trabajo muy cómo, robusto y con una comunidad muy amplia a sus espaldas. Se puede decir

que ofrece que el sistema desarrollado sea flexible, escalable y robusto. Además, ofrece unas plantillas en las que basar todo el proyecto muy intuitivas y fáciles de usar al crear las soluciones para desarrollar cualquier proyecto promoviendo el uso de patrones de diseño como puede ser la inversión de control utilizando la inyección de dependencias que ayuda a crear soluciones más eficaces y mejor diseñadas para la etapa de realización de pruebas.

Para la realización de la Web Api⁹ se ha seguido las directrices de la filosofía de Api Rest utilizando las principales operaciones para el acceso de datos a través de peticiones HTTP a través de las interfaces que nos proporciona la Api en cuestión. Las principales operaciones que soporta las Api Rest son:

- Post: para las operaciones de creación de nuevos elementos en la tabla/colección
- Put: para las operaciones de actualización de algún dato en concreto o del objeto entero en cuestión.
- Get: para las operaciones de acceso y consulta a los datos que están almacenados en la base de datos.
- Delete: para las operaciones de borrado de cualquier dato almacenado en la base de datos.

Experiencia

Este marco de trabajo aporta al desarrollador un desarrollo más seguro y dinámico ya que no tiene una curva de aprendizaje muy inclinada y al tener una comunidad muy extendida se puede encontrar soluciones a cualquier duda de forma sencilla. Los paquetes Nugets o librerías que la apoyan son de verdadera utilidad como puede ser el uso de LinQ¹⁰, una librería que aporta el uso de expresiones lambda y métodos muy utilizados por los desarrolladores y hacen la vida más fácil una vez se conocen los conceptos de la programación orientada a objetos (OOP). El lenguaje utilizado para el desarrollo en este entorno es C#¹¹ el cual tiene muchas similitudes con Java. Para la realización de este proyecto y la construcción de la Web Api se han utilizados algunos paquetes Nugets como pueden ser Swagger, JWT¹² para la gestión de los tokens de acceso y la securización de la Api, Newtonsoft¹³ que provee una utilización de los documentos Json de una manera más sencilla y cómoda y el conector de Mongo para la conexión a la base de datos de CosmosDB de Azure.

Visual Studio Code¹⁴

Definición

Es un editor de código fuente de código abierto desarrollado por Microsoft para los sistemas operativos de Windows, MacOS y Linux. Es gratuito y se puede tener cabida una gran personalización para adaptarse a las necesidades del desarrollador.

Motivación

La principal motivación para usar este entorno de desarrollo tan sencillo, pero a la par complejo, ha sido aprender a desarrollar un proyecto desde Visual Studio Code con el fin de ampliar el rango de conocimientos con un entorno distinto a Visual Studio 2017.

Experiencia

La adaptación a este nuevo entorno de desarrollo si se procede desde Visual Studio 2017 prácticamente es de algunos días para llegar a comprender que este entorno es modular pudiendo ampliar las funcionalidades que aporta a partir de extensiones desarrolladas por personas que deciden contribuir a mejorar este editor de código fuente o de empresas que deciden sumarse a la causa. Entre otras extensiones cabe destacar Git Lens para poder interactuar con el repositorio de código fuente de Azure DevOps, la extensión de Angular la cual aporta atajos y mejoras a la hora de programar y desarrollar un código de mayor calidad y otras tantas extensiones de gran utilidad.

Angular¹⁵

Definición

Angular es un marco de trabajo de código abierto desarrollado en TypeScript¹⁶ y mantenido por Google. El principal fin que tiene este marco de trabajo es crear aplicaciones web y aumentar el número de aplicaciones basadas en patrón de diseño MVC¹⁷, todo este esfuerzo tiene el fin de que el desarrollo y la etapa de pruebas sea más sencilla y dinámica.

Motivación

En un primer momento se debatió en que tecnología se debía elegir para la parte frontal del proyecto y entre las opciones a elegir se encontraban React,

Angular y VueJs¹⁸. La última opción se descartó ya que es un framework muy reciente, la primera reléase estable ha salido en el mes de marzo de 2019, y aunque en los últimos meses está cogiendo mucho terreno a los otros dos gigantes no acababa de convencer en el momento de decisión. El gran debate se estableció en si el proyecto debía ser desarrollado con React o Angular, hubo un tiempo de investigación y asimilación de información para optar por uno ya que no se conocía ninguno de los dos. Se buscó información sobre la curva de aprendizaje de ambos marcos de trabajo y, se llegó a conclusión de que Angular era un marco de trabajo más gradual para los desarrolladores novatos y el aprendizaje era de forma más paulatina. Además, unas de las cosas que decantó la balanza fue que el lenguaje de programación con que se desarrolla Angular que es Typescript, un superconjunto de JavaScript, aportando multitud de características a este famoso lenguaje el cual no se poseían conocimientos. Typescript ayuda mucho a la hora de desarrollar de forma más segura y ágil ya que detecta muchos errores en tiempo de compilación.

Experiencia

Una de las causas por las que se eligió Angular fue por usar Typescript a la hora de desarrollar, ya que este superconjunto de JavaScript añade clases, interfaces, tipados en las variables y un largo etcétera. Este lenguaje fue familiar ya que es orientado a objetos como lo es el lenguaje usado en la parte del back-end como es C#. Angular posee una terminal la cual es de gran ayuda a la hora de crear cualquier elemento propio de este marco de trabajo y conocer los comandos básicos puede ahorrar tiempo y esfuerzos. La forma de desarrollar es relativamente sencilla de comprender ya que se basa en módulos, dichos módulos se conocen como componentes. Cada componente se divide en varios archivos como un HTML donde se maqueta la vista, un archivo Typescript que hace la función de controlador y un CSS para dar estilos, en el caso de este proyecto. Al ser un marco de trabajo tan modular se pueden acoplar módulos externos desarrollados por otras personas/equipos, las fuentes más fiables donde se han encontrados estos componentes externos son GitHub, NPM¹⁹ y páginas de empresas dedicadas al desarrollo de componentes para marcos de trabajo como Angular. Algunas de estas empresas tienen los componentes subidos a sus propias páginas web oficiales. Para la instalación y gestión de

estos paquetes se utiliza NodeJs²⁰. En este proyecto se ha tenido especial consideración a la hora de elegir componentes externos ya que se pueden usar componentes que no estén diseñados y desarrollados de una forma eficiente y optima y, además el dueño de dicho componente no esté dando el soporte y actualizaciones adecuadas para las posibles nuevas versiones futuras de Angular. Los componentes externos permiten que desarrollos que se suelen repetir en todas las aplicaciones web no los tenga que desarrollar de nuevo cada desarrollador sino usar ese componente externo, haciendo el desarrollo mucho más simple y ágil. Es cierto que no siempre el componente externo se adapta a la perfección a los requerimientos necesarios y se debe de modificar para que funcione y se adapte correctamente. Los componentes externos²¹ utilizados por en este proyecto son:

- ngx-quill
- ng-Bootstrap
- ngx-Bootstrap
- ng2-Select
- Bootstrap
- Docx
- ng2-validation
- file-saver
- ng5-slider
- Font-awesome

Al elegir estos componentes externos se ha puesto especial atención en que estuviesen desarrollados por equipos o desarrolladores con apoyo de la comunidad y con visión de seguir manteniendo dichas librerías en futuras versiones. Este proyecto usa Angular 5, ya que en el momento de crear dicho proyecto la versión sexta del marco de trabajo acababa de salir y se decidió montar la plataforma sobre la versión más estable.

5

Diseño del modelo de datos

En este apartado se abordarán las elecciones tomadas en cuanto al modelo de datos que se va a seguir para el desarrollo del proyecto.

Antes de ver el modelo de datos en profundidad se va a explicar cómo es la arquitectura construida para que el frontend se comuniquen con la base de datos, para que esta pueda ser explotada de forma segura y sin complicaciones.

Para ello se ha decidido construir una API Rest la cual es la encargada de ofrecer la persistencia a los datos almacenados en Azure CosmosDB.

Para el acceso a los datos se han creado unos modelos equivalentes tanto en la parte front-end como en el back-end, en Typescript y en C# respectivamente.

Estos modelos se van a explicar con más detenimiento con la ayuda de Swagger, el cual proporciona una vista sencilla de los modelos de datos de la Api, al hacer un mapeo directo de los modelos de datos creados en el back-end. Todos los atributos referentes a identificadores únicos en los modelos tienen un tipado el cual se manejará una cadena de texto, más conocido como *string*, ya que para las peticiones a la Api es mucho más sencillo y menos problemático. Además, la gestión de los identificadores para la relación de documentos se está haciendo de forma manual y no se usan los identificadores únicos que genera internamente la base de datos.

Profundizando en los modelos usados para el correcto funcionamiento de la plataforma:

- DictionaryWord

```
DictionaryWord {  
  id          string  
  title       string  
  description  string  
}
```

Este modelo es usado para las palabras que puede almacenar el escritor. Dicho modelo está incrustado en el modelo de *Writer* en forma de una lista como se ha

Figura 8 – Modelo de dictionaryWord
podido observar.

- Writer

```
Writer {
  id string
  writerId string
  fullName string
  city string
  country string
  birthdate string
  email string
  password string
  alias string
  category string
  monthsExperience integer($int32)
  salt string
  refreshToken string
  urlImgProfile string
  createdOn string
  biography string
  dictionary > [...]
  taskslist > [...]
}
```

En este modelo se puede ver las propiedades que va a tener un escritor cuando se registre en el sistema. Esta información básica como puede ser el email con el que se registra, contraseña, nombre completo, alias entre otras. Toda esta información no es introducida por el usuario, sino que se autogeneran ya que se requieren en ciertas partes de

Figura 9 – Modelo de writer

la plataforma para una mejor navegación a través de ella. Estos campos no introducidos por el escritor son el refreshToken y el Salt, que se utilizan para la gestión de los tokens de acceso y para añadir aleatoriedad a la contraseña para una mayor seguridad respectivamente. Adicionalmente este modelo contiene dos listas de objetos de otros modelos como son la lista de palabras del diccionario y la lista de tareas, estos dos modelos se explicarán a continuación.

- TaskWriter

```
TaskWriter {
  id string
  title string
  description string
  deadline string
  state string
  completedon string
}
```

Este modelo hace referencia a las tareas que el usuario podrá gestionar en el apartado del planificador. Este modelo tiene las propiedades básicas y necesarias para que el usuario pueda organizarse creando tareas. Algunas de sus propiedades pueden ser el título,

Figura 10 – Modelo de taskWriter

descripción, el estado en el que se encuentra dicha tarea, la fecha límite que el escritor indica y la fecha de cierre que se informa una vez se cierra la tarea. Este modelo está incrustado en el modelo de *Writer* en forma de lista.

- Book

```
Book ▾ {  
  id                string  
  bookId            string  
  writerId          string  
  title             string  
  imageUrl          string  
  description       string  
  generer           string  
  wordsNumber       string  
  updateOn          string  
  chapters          > [...]  
  glossaryWords     > [...]  
  notes             > [...]  
}
```

En este modelo se puede observar las propiedades que tendrá un libro al ser creado en el sistema. Los campos `bookId`, `writerId`, `wordsNumber`, `updateOn`, e `imageUrl` son generados y gestionados por el propio sistema. El campo `wordsNumber` representa un contador de palabras del libro completo, `updateOn` la fecha de actualización del libro, e `imageUrl` almacena la ruta de la imagen

Figura 11 – Modelo de book

de portada. Los campos restantes de tipo `string` se obtienen a raíz de la inserción por parte del usuario.

Por la simplicidad que presentan y por la relación funcional que mantienen con un libro, también se incluyen tres subconjuntos en el modelo de *Book*: `chapters`, `glossaryWords`, y `notes`.

- GlossaryWord

```
GlossaryWord ▾ {  
  id                string  
  title             string  
  description       string  
}
```

Este modelo refleja las palabras que el escritor puede almacenar para explicar y dar significado propio a una serie de palabras que aparezcan en cualquiera de sus obras. Esta lista de palabras del

Figura 12 – Modelo de glossaryWord

glosario están incrustadas en el modelo de *Book*, es decir, cada libro contiene su propio glosario.

- Note

```
Note ▾ {  
  id          string  
  title       string  
  description  string  
}
```

Figura 13 – Modelo de note

Este modelo refleja las notas que puede realizar el escritor en cada libro. Estas notas están compuestas por un título y una descripción donde se explicará algún suceso que decida el propio escritor. Este

modelo está incrustado en forma de lista en el modelo de Book, es decir, cada libro contiene su propio glosario.

- SentimentsAnalytic

```
SentimentsAnalytic ▾ {  
  analisisId    string  
  writerId     string  
  bookId       string  
  sentimentAnalysis > [...]  
  updatedOn    string  
}
```

Figura 14 – Modelo de sentimentsAnalytic

Este modelo hace referencia al análisis de sentimientos que el escritor tiene la opción de realizar una vez que ha escrito algún capítulo de un libro. La respuesta a la petición del análisis se

almacenará en el atributo *sentimentAnalysis* el cual contendrá toda la información relevante al análisis realizado. Las otras propiedades que se pueden observar son internas para saber a qué escritor y libro pertenece dicho análisis.

- Chapter

```
Chapter ▾ {  
  chapterId    string  
  title        string  
  content      string  
  text         string  
  wordsNumber  string  
  updateOn    string  
  screenPlayEvents > [...]  
}
```

Figura 15 – Modelo de chapter

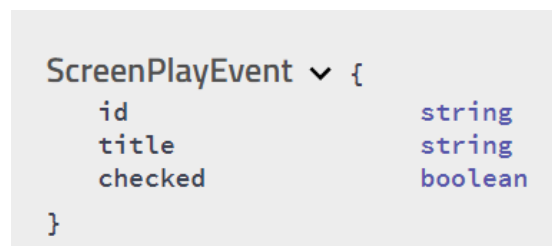
El modelo Chapter, representa el capítulo contenido en el libro. Sus campos reciben la información a partir del escritor, durante la creación de libro, o durante la propia escritura. ChapterId, wordsNumber, y updateOn son generados por el sistema.

En esta ocasión, wordsNumber representa el número de palabras del capítulo. La suma de todos los números de palabras, de todos los capítulos de un libro,

genera el número de palabras contenido en wordsNumber del libro. Content y text almacenan el cuerpo del capítulo, con la diferencia de que el primero mantiene el formato del editor, y el segundo es tan solo texto plano.

Además, mantiene un subconjunto denominado screenPlayEvents, que supone ni más ni menos que los eventos del guion rápido de cada capítulo.

- ScreenPlayEvent



Este modelo se alimentará cada vez que un escritor introduzca nuevos sucesos o eventos que se producen durante el desarrollo del capítulo. El campo checked hace referencia a si el

Figura 16 – Modelo de screenPlayEvent

evento ha sido completado a lo largo del capítulo, o bien está pendiente a realizar.

- TextAnalytic

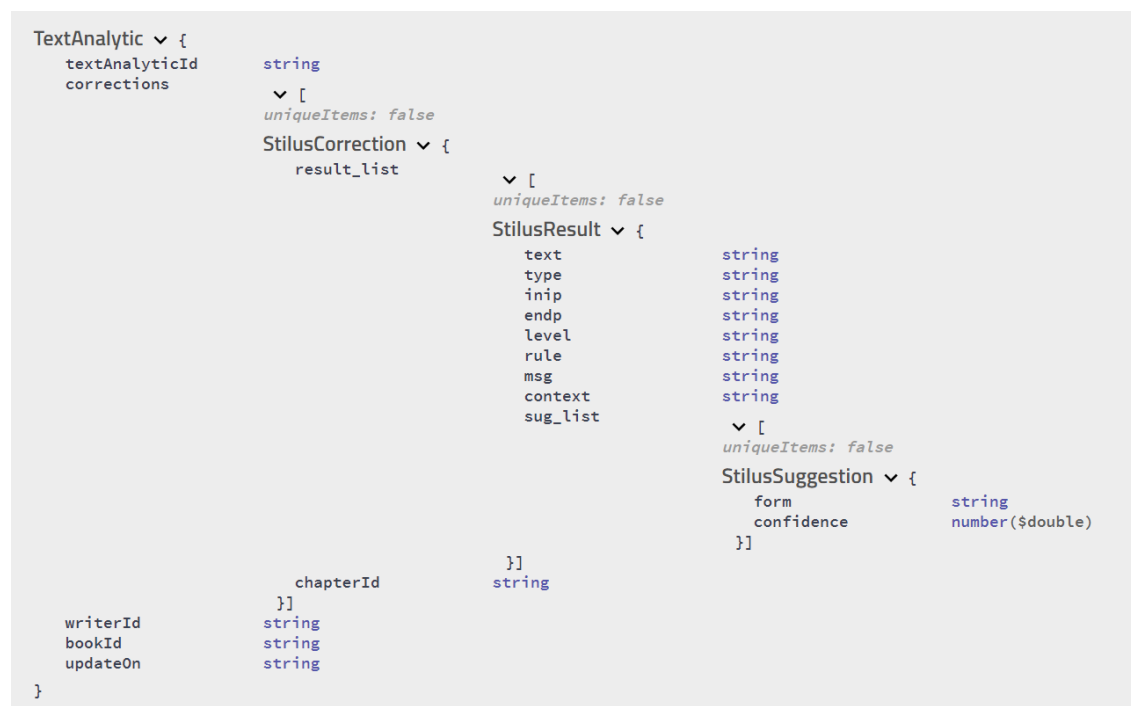


Figura 17 – Modelo de textAnalytic

En esta imagen se puede apreciar el modelo de *TextAnalytic*, almacenamiento de los análisis de texto realizados para la corrección de faltas de ortografía.

Es un modelo pensado para que cada libro tenga un solo análisis, creado una única vez, que se irá actualizando en las siguientes peticiones de análisis. Mantiene su identificador único, pero se vale de los identificadores de escritor y libro para comprobar la existencia de análisis previo.

Así mismo, el subconjunto *corrections* mantiene el identificador del capítulo del que proceden las faltas de ortografía. Este subconjunto es el mayor grueso del modelo, pues representa la respuesta generada por la API de *Stilus*, la cual dota al sistema de información realmente interesante.

Cabe destacar el texto erróneo (ya sea una frase o palabra), el mensaje explicativo del error, y el conjunto de sugerencias para resolverlo.

- Character



```
Character {
  id: string
  characterId: string
  writerId: string
  bookId: string
  urlImgCharacter: string
  fullName: string
  minAge: integer($int32)
  maxAge: integer($int32)
  genre: string
  birthPlace: string
  sexOrientation: string
  rol: string
  aliases: > [...]
  ancestors: > [...]
  couples: > [...]
  sons: > [...]
  brothers: > [...]
  biography: string
  complexion: string
  bodyType: string
  height: integer($int32)
  toneVoice: string
  volumeVoice: string
  clothesType: string
  skinTone: string
  skinType: string
  hairColor: string
  hairSize: string
  bodyHair: string
  eyesColor: string
  eyesType: string
  noseType: string
  earsType: string
  gestures: string
  motivations: string
  manies: string
  hobbies: string
  behaviour: string
  convictions: string
  fears: string
  modifiedOn: string
}
```

El modelo de personajes refleja toda la información que necesita el escritor para moldear a los personajes que van a dar vida a las obras, por lo tanto, necesita almacenar una serie de propiedades y características propias de cada personaje. Todas estas propiedades las insertará el escritor para dar personalidad y forma a los personajes que el decida, y podrá consultarlos en la galería de personajes en todo momento.

Figura 18 – Modelo de character

6

Desarrollo de la aplicación

En este apartado se explicarán de forma detallada como se han desarrollado los requisitos funcionales que competen y se especificaron en el apartado correspondiente. Para el desarrollo de la aplicación se han seguido las directrices que especifican los requisitos funcionales y sus casos de uso respectivos, donde se explican qué funcionalidad se quiere llevar a cabo y que objetivo se aborda para que el sistema cumpla con los objetivos.

Registro del usuario en la aplicación

Para cubrir dicha funcionalidad básica la plataforma debe proporcionar al usuario, que aún no está inscrito en el sistema, un formulario web donde poder registrarse rellenando una serie de campos necesarios. Dicho formulario será accesible desde la pantalla de inicio de la aplicación.



Figura 19 – Página inicial

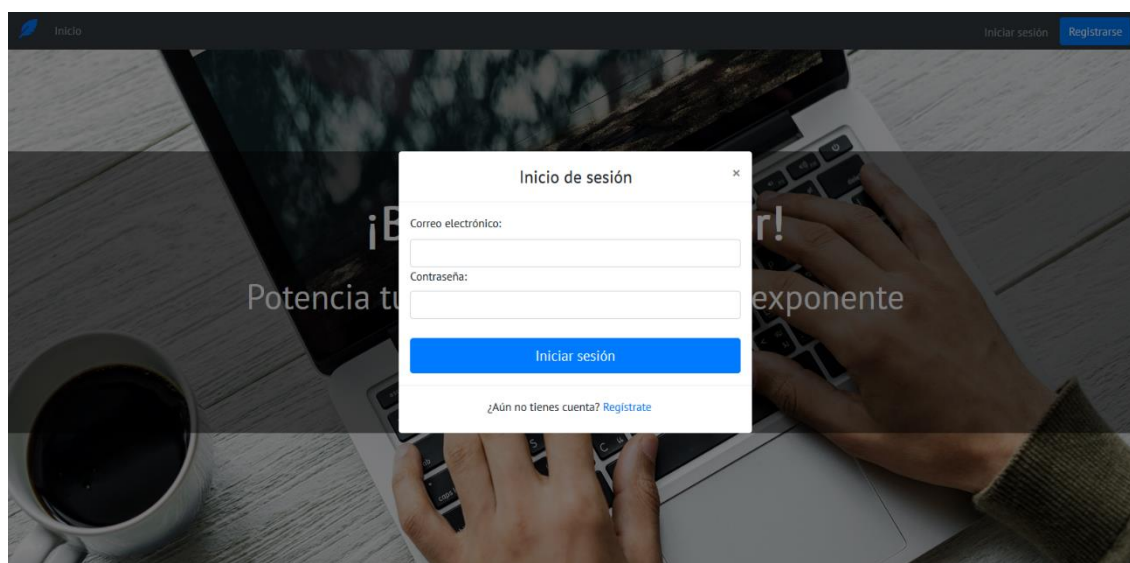


Figura 20 – Página de inicio de sesión

El formulario será accesible desde tres enlaces:

- 1- Desde la barra de navegación cuando no se ha iniciado sesión.
- 2- Desde la pantalla de bienvenida.
- 3- Desde la pantalla de inicio de sesión.


En ambos casos se redirigirá a una pantalla exclusiva donde se presenta el formulario en cuestión a rellenar. El formulario contiene una serie de parámetros a rellenar por el usuario que desea registrarse en la aplicación.

Figura 21 – Página de registro de usuario

Todos los campos son requeridos obligatoriamente para poder enviar el formulario de registro y dar de alta al nuevo usuario en el sistema. Cada uno de los campos tiene una serie de restricciones para asegurar la validez de los datos

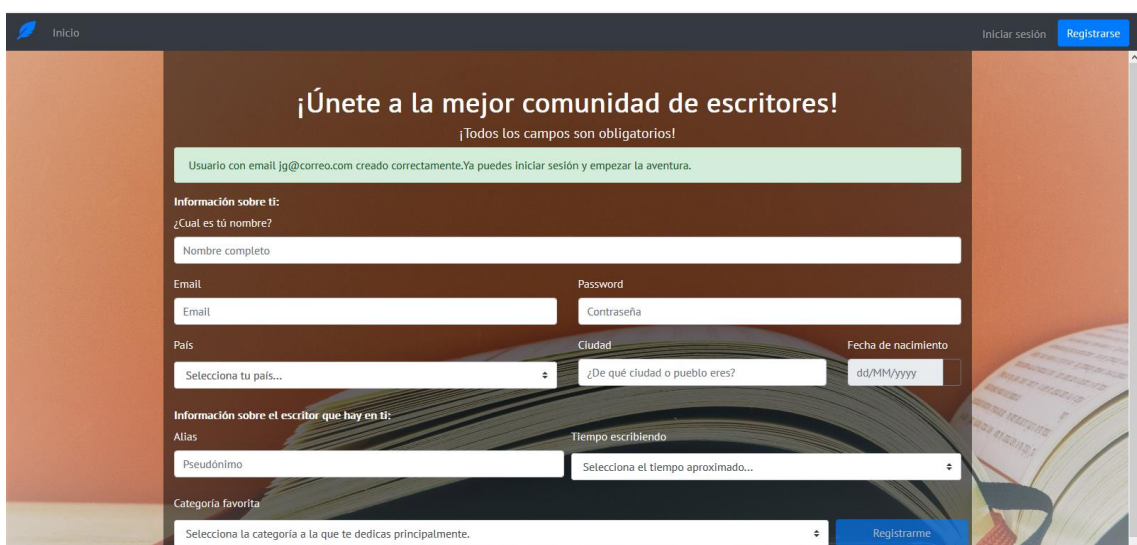
por parte del usuario. Además, el botón de envío del formulario aparecerá desactivado mientras todos los campos no estén informados correctamente, una vez se cumplan las restricciones el botón se activa y el usuario podrá enviar el formulario.

En el caso de que no se rellene de forma adecuada los campos requeridos aparecerá un texto de alerta bajo cada campo indicando la restricción que no se está cumpliendo para que el usuario pueda corregir los datos introducidos. A su vez cuando los datos son correctos y el usuario se registra, aparecerá un cuadro de texto al inicio de la página indicando el resultado del registro donde el usuario podrá ver si el registro ha sido satisfactorio o ha ocurrido algún inconveniente.



The screenshot shows a registration form titled "¡Únete a la mejor comunidad de escritores!". The form includes fields for "Nombre completo", "Email", "Password", "País", "Ciudad", "Fecha de nacimiento", "Pseudónimo", "Tiempo escribiendo", and "Categoría favorita". Each field has a red border and a red error message below it: "El nombre es un campo obligatorio.", "El email es un campo obligatorio.", "La contraseña es un campo obligatorio.", "Selecciona tu país...", "¿De qué ciudad o pueblo eres?", "dd/MM/yyyy", "Pseudónimo", "Selecciona el tiempo aproximado...", and "Selecciona la categoría a la que te dedicas principalmente.". A green success message at the top says: "Usuario con email jg@correo.com creado correctamente. Ya puedes iniciar sesión y empezar la aventura."

Figura 22 – Página de registro con advertencias



The screenshot shows the same registration form as Figure 22, but with a green success message at the top: "Usuario con email jg@correo.com creado correctamente. Ya puedes iniciar sesión y empezar la aventura." The "Registrarme" button is now active and blue.

Figura 23 – Página de registro con confirmación

Los datos pedidos al usuario para darse de alta en el sistema se almacenarán en la base de datos. Se sabe que hay ciertos datos sensibles que deben protegerse ante posibles fallos de seguridad, por lo tanto, se ha decidido que en todo momento las contraseñas se guarden de manera que, si se comprometiese la seguridad del sistema o la base de datos, los datos sensibles de los usuarios no estuviesen, en texto plano, expuestos y comprometidos.

Para ello el método empleado para encriptar la contraseña que se va a utilizar es el algoritmo de encriptación HMACSHA-512²². Esta función hash que se construye a partir de SHA-512 perteneciente a la familia de SHA-2. En este caso se va a configurar con unas 10000 iteraciones y finalmente se obtendrá una clave derivada de 64 bytes, es decir, 512 bits.

Al recibir el campo de texto de la contraseña del usuario se va a generar aleatoriamente una cadena de caracteres extra que se añadirá al final de la contraseña introducida. Dicha cadena la función principal que tiene es añadir aleatoriedad a la contraseña, comúnmente se le conoce con el nombre de *salt*. En este caso el *salt* se genera cada vez que un usuario se registra, por lo tanto, es único para cada usuario. Esto tiene como aliciente una mayor complicación para descifrar las contraseñas en caso de que un atacante entre en la base de datos.

Dicho salt al ser único para cada usuario se guardará en la base de datos con el resto de los datos del usuario. El por qué se guarda en la base de datos es para que cuando el usuario intente iniciar sesión se concatenará el salt del usuario a la contraseña introducida en el intento de inicio de sesión y se volverá a encriptar comparándose si las cadenas encriptadas coinciden, en ese caso de ser la misma el usuario que estaba intentando iniciar sesión podrá acceder a la plataforma.

Inicio de sesión del escritor

Para cubrir dicha funcionalidad la plataforma proporcionará al escritor un formulario web donde poder iniciar sesión y acceder a todas las funcionalidades que ofrece la plataforma. Dicho formulario será accesible desde la barra de navegación superior a través de un botón el cual al pinchar sobre él hará que

aparezca un modal con dicho formulario. El formulario de inicio de sesión está compuesto por dos campos:

- Email con el que se ha registrado el usuario.
- La contraseña con la que se ha registrado el usuario.

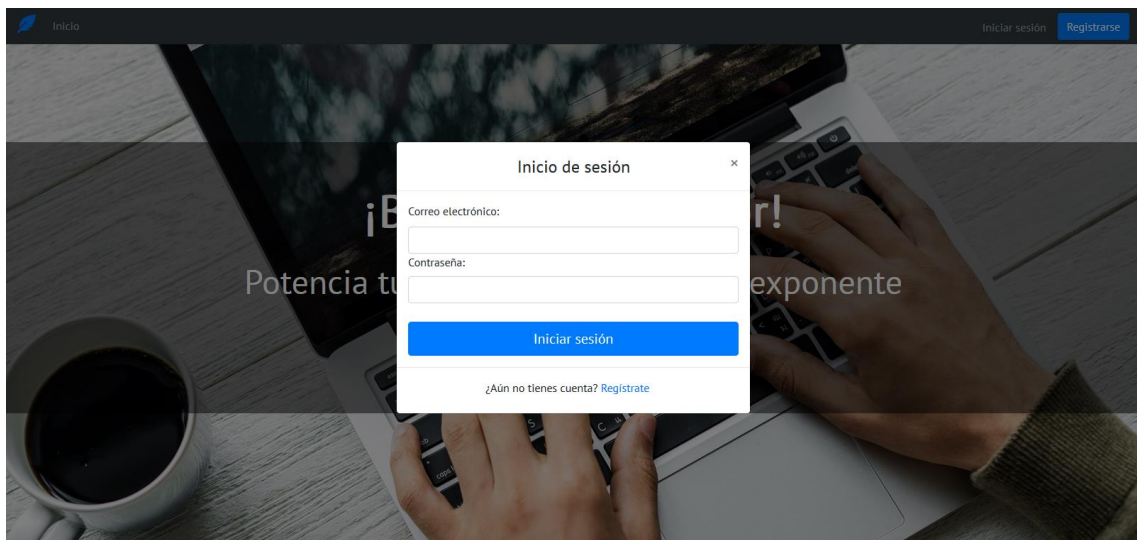


Figura 24 – Página de inicio de sesión

Al rellenar los campos que pide el formulario de inicio de sesión y pinchar el botón para iniciar sesión, se comprobará que el email y la contraseña introducidas existen en la base de datos. En el caso de que algunos de los dos

A screenshot of the same login modal as in Figure 24, but now showing an error message. The modal is titled 'Inicio de sesión' with a close button (X) in the top right corner. A red error message box at the top says 'Usuario o contraseña inválida.' Below this, the 'Correo electrónico:' field contains the text 'jg@correo.com'. The 'Contraseña:' field is filled with dots. A blue button labeled 'Iniciar sesión' is at the bottom. Below the modal, there is a link that says '¿Aún no tienes cuenta? Regístrate'.

Figura 25 – Inicio sesión erróneo con advertencia

campos no sean correctos se le notificará al usuario de que los campos introducidos son incorrectos para que revise y vuelva a intentarlo.

En cambio, si las credenciales son correctas se habilitarán algunos enlaces en la barra superior de navegación para poder acceder a diferentes partes de la plataforma como el propio perfil del escritor, el planificador de tareas, diseño de hoja de personajes entre otras características y se redirigirá al perfil del escritor.

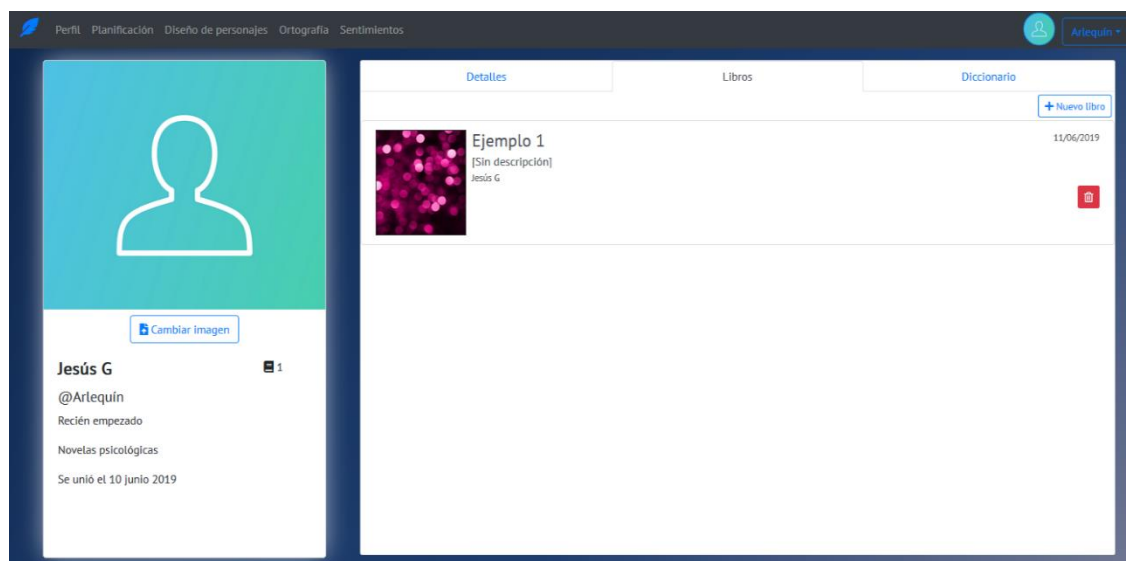


Figura 26 – Página de perfil del escritor

Como se ha visto anteriormente, la contraseña de los escritores no se almacena en la base de datos directamente como texto plano, sino que se encripta, por lo tanto, a la hora de comprobar que las credenciales son correctas se comprueba si el hash aplicado a la contraseña introducida en el formulario de inicio de sesión coincide con el hash almacenado al usuario con el email introducido en el formulario de inicio de sesión.

Tokens de acceso

Para la gestión de los usuarios se ha decidido usar tokens de acceso haciendo que toda la gestión sea transparente al usuario que utiliza la plataforma.

Para la gestión de los tokens se ha decidido utilizar el estándar Json Web Tokens, más conocidos como tokens JWT.

Con esto se añade un procedimiento más complejo para el inicio de sesión y la autorización de dicho escritor a las en la plataforma. Dichos tokens entran en

juego cuando el escritor intenta acceder al sistema. Cuando el escritor ha iniciado sesión, se comprueba que el email y la contraseña sean correctas, es decir, cuando inicia sesión al escritor se le asigna un token de acceso y un token de refresco y a partir de este punto toda acción que realice el escritor en la plataforma va acompañada de los tokens generados, para que así el sistema no tenga que comprobar todo el tiempo si el email y contraseña son válidos.

Como el token de acceso por motivos de seguridad tiene un tiempo de validez, es posible su configuración, se crea un segundo token para poder obtener uno nuevo en el caso de que caduque. El token de refresco no es nada más que un identificador único generado aleatoriamente que se renueva cada vez que el escritor inicia sesión o el sistema genera un token de acceso nuevo, y es almacena en la base de datos.

El verdadero token JWT es el token de acceso que no se almacenará en ningún momento en la base de datos. Este token se guarda en el almacenamiento local del navegador para que la aplicación pueda consultarlos de manera rápida, y se eliminarán cuando el escritor decida cerrar la sesión o estos caduquen y la aplicación los elimine.

Todo esto aporta una experiencia al escritor muy cómoda ya que si inicia sesión una vez y el token de acceso le caduca, no tendrá que volver a introducir el email y contraseña por que la aplicación se encarga de generar un nuevo token, con el token de refresco que está en el almacenamiento local del navegador y la base de datos. Si dicho token de refresco coincide en los 2 lugares donde se ha almacenado se le generará un nuevo token de acceso y un nuevo token de refresco, por lo tanto, todo este proceso está ocurriendo por detrás del escritor que no es consciente de todo este proceso y puede seguir en la aplicación sin tener que volver a identificarse.

Toda esta lógica se ha implementado con un middleware propio de Angular que recibe el nombre de interceptores. Esto permite definir cierta lógica en dicho componente, el interceptor, que va a capturar toda petición que se realice a través de HTTP y aplica ciertas reglas añadiendo cabeceras, realizando comprobaciones antes de dejar pasar la petición o incluso abortando el envío. Por lo tanto, en el interceptor se han definido ciertas reglas de verificación de los tokens de acceso, si el token es correcto se introduce en la cabecera de la

petición para poder acceder a las APIs securizadas. En el caso de que el token esté caducado, el interceptor se encargará de que, si el token de acceso y el token de refresco son correctos y, pertenecen al escritor, generará unos tokens nuevos añadiéndolos a la cabecera poder acceder a las Apis y que el escritor pueda navegar sin tener que volver a iniciar sesión.

En el caso de que los tokens sean corruptos por cualquier razón o no exista ningún token en el almacenamiento local del navegador, el interceptor se encargará de eliminar los datos del almacenamiento local del navegador y cerrando la sesión del escritor.

Cierre de sesión del escritor

Para cubrir dicha funcionalidad la plataforma proporcionará al escritor mecanismos para borrar la información relativa al usuario que se guarda desde que inicia sesión. Para ello dispondrá de un botón alojado en la barra superior de navegación con el cual podrá cerrar la sesión que tiene abierta.

Este no es el único mecanismo que tiene la aplicación para cerrar la sesión del escritor, como se ha visto en el inicio de sesión, como se utilizan tokens para la administración de la sesión del escritor, puede ocurrir que el token sea corrupto o haya caducado, es decir, si el token es corrupto por cualquier razón la aplicación directamente borrará los datos que se hayan almacenado en el almacenamiento local del navegador y los servicios, cerrando la sesión.

En el otro caso en el que el token es válido, pero se ha caducado puede ocurrir el siguiente escenario:

El interceptor, ya comentado antes, capturará la petición HTTP validando si el token es válido pero caducado y el token de refresco almacenado en el almacenamiento local del navegador y en la base de datos coinciden.

Si los tokens de refresco no coinciden entonces no se le generará un nuevo token de acceso para el escritor y la sesión debe cerrarse por seguridad.

Este escenario ocurre por el simple hecho de que solo estamos almacenando un token de refresco al iniciar sesión para cada escritor, sobrescribiendo el anterior token de refresco. Esto da pie a que el escritor solo pueda tener una sesión abierta simultáneamente ya que sino los tokens de refresco difieren y la aplicación se responsabilizará de cerrar su sesión.

Esto ocurre si el escritor accede iniciando sesión desde un ordenador personal y en otro momento por cualquier motivo vuelve a iniciar sesión desde otro ordenador diferente al primero, teniendo la sesión abierta en el primero. Por lo tanto, cuando regrese al ordenador personal la aplicación cerrará la sesión abierta al no coincidir los tokens de refresco y el escritor deberá identificarse con sus credenciales de nuevo.

Perfil del escritor

El escritor nada más iniciar sesión accederá a una sección donde convergen la gran parte de las funcionalidades de la plataforma, esta sección es el perfil del escritor. Al iniciar sesión se habilitan todas las secciones accesibles desde la barra de navegación, pero es solo desde el perfil del escritor donde podrá ver todas las obras y acceder a cada una de ellas.

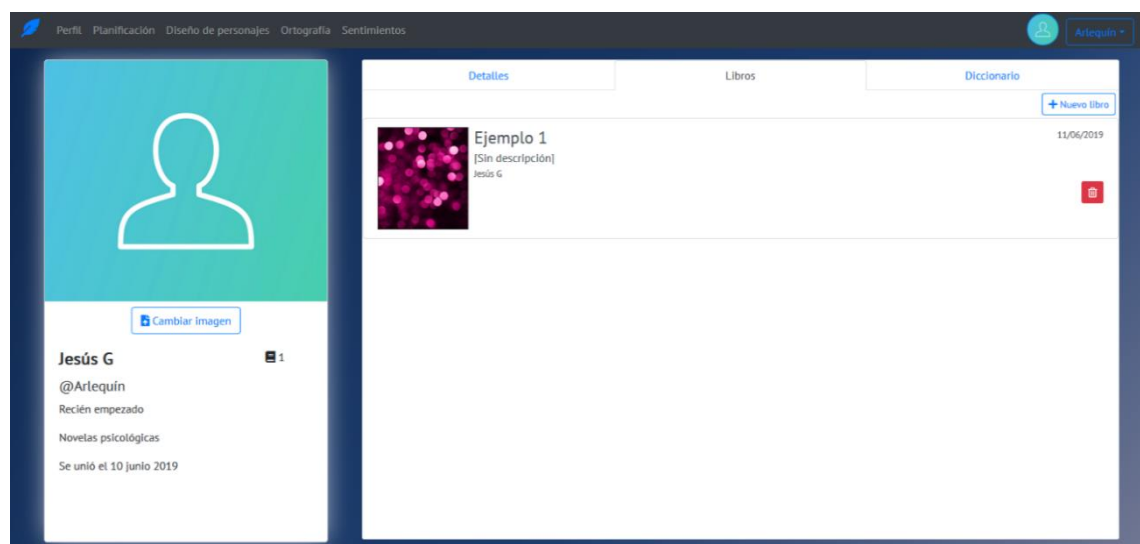


Figura 27 – Página del perfil del escritor

En la parte izquierda de la sección se puede observar un resumen de la información del escritor, donde puede cambiar la foto del perfil que desea que aparezca.

Como se puede observar, el escritor podrá ver a simple vista el listado de sus obras que está desarrollando y en cualquier momento acceder a ellas. Así mismo, podrá crear y añadir nuevas obras o en caso contrario eliminar las obras que no desee guardar.

Al lado de la pestaña del listado de *libros* encontramos otra pestaña *detalles* la cual el escritor puede acceder y modificar su información básica, además puede añadir una pequeña biografía en la que escribir su historia.

Figura 28 – Página de inicio de sesión

Además de las pestañas de *detalles* y *libros* se puede ver otra pestaña donde indica diccionario, esta pestaña será explicada en el siguiente apartado.

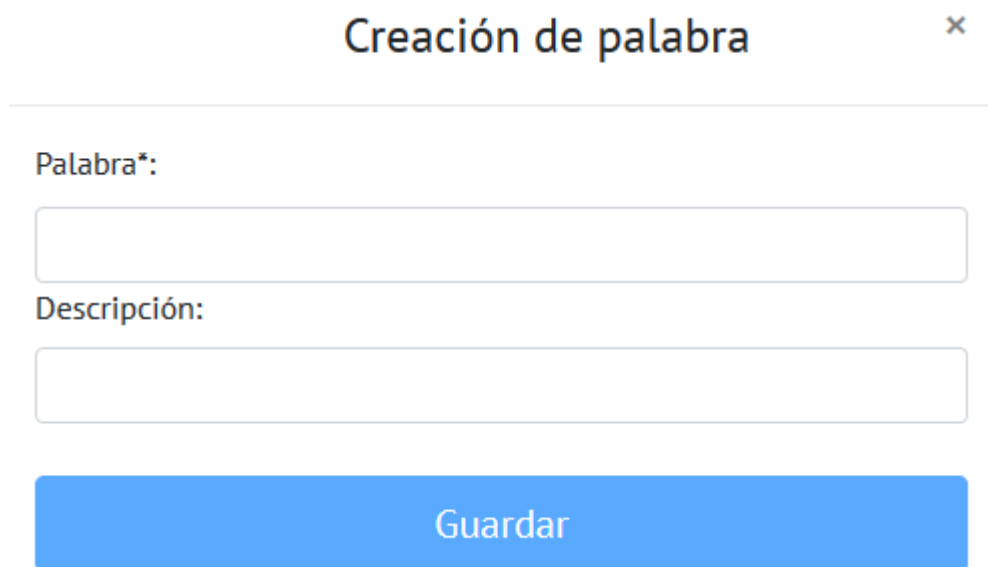
El escritor tendrá un diccionario personal

En el propio perfil del escritor, este podrá acceder a la pestaña *diccionario* y podrá ver todas las palabras que el propio escritor ha añadido por alguna razón la cual creía que esas palabras serían de utilidad en su futuro.

Figura 29 – Página de diccionario del escritor

El listado de palabras del diccionario se muestra en forma de acordeón, el cual muestra solo los títulos de las palabras, en el caso de que se quiera ver el significado el escritor deberá pinchar sobre la palabra y se desplegará la descripción.

Este listado está alimentado por el propio escritor, por lo tanto, tiene accesible a simple vista un botón para poder añadir nuevas palabras al listado. Al pinchar sobre el botón aparecerá un simple formulario el cual debe rellenar el escritor.

El formulario se titula "Creación de palabra" y tiene un icono de cerrar (X) en la esquina superior derecha. Contiene dos campos de entrada: "Palabra*" y "Descripción:". Debajo de estos campos hay un botón azul con el texto "Guardar".

Creación de palabra

Palabra*:

Descripción:

Guardar

Figura 30 – Formulario de creación de palabra

El escritor solo deberá rellenar dichos campos y pinchar el botón de guardar.

El formulario acepta que solo se rellene la palabra, en cambio si no se rellena el primer campo aparecerá un mensaje de advertencia.

Creación de palabra

×

El título es un campo obligatorio.

Palabra*:

Descripción:

Guardar

Figura 31 – Formulario de nueva palabra erróneo con advertencia

Así mismo el escritor podrá modificar las palabras ya creadas en el listado o eliminarlas. En el caso de la modificación, aparecerá el mismo formulario que en la creación con la información cargada de la palabra seleccionada.

Edición de palabra

×

Palabra*:

Amover

Descripción:

Destituir, deponer a alguien de su empleo o destino

Guardar

Figura 32 – Formulario de modificación de palabra del diccionario

Y en el caso de querer eliminar alguna palabra del listado, el sistema pedirá confirmación de borrado.

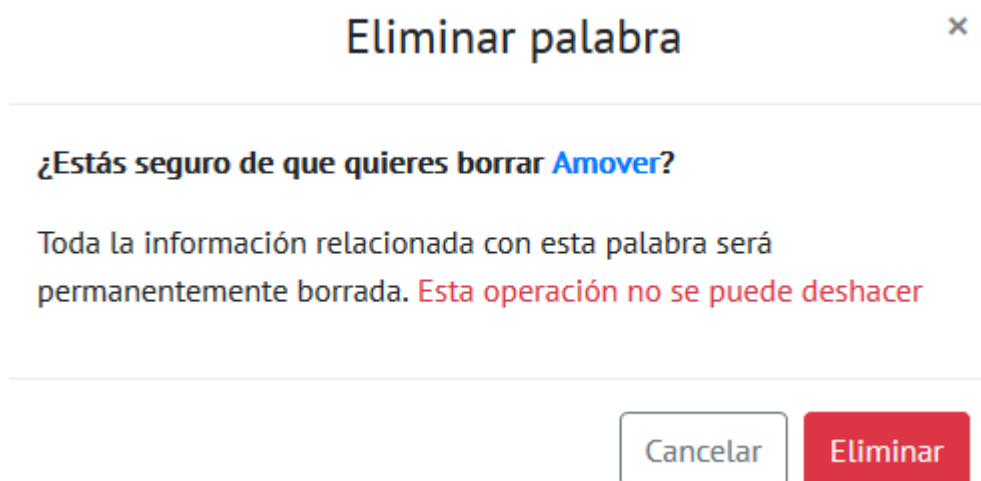


Figura 33 – Página de confirmación de borrado de palabra del diccionario

Cada libro tiene un glosario asociado

El escritor por cada libro que crea tiene acceso a un glosario propio de cada libro en el que añadir palabras. Estas palabras que se definen en el glosario de cada libro pueden que sean palabras polisémicas y el escritor quiera aclarar el significado que tendrá en dicha obra o, que el escritor haya inventado palabras y desee definir las para que se entiendan en el contexto del libro.

Para acceder al glosario de cualquier libro se puede hacer accediendo al libro desde el perfil del escritor, y pinchando sobre la pestaña de *glosario*.

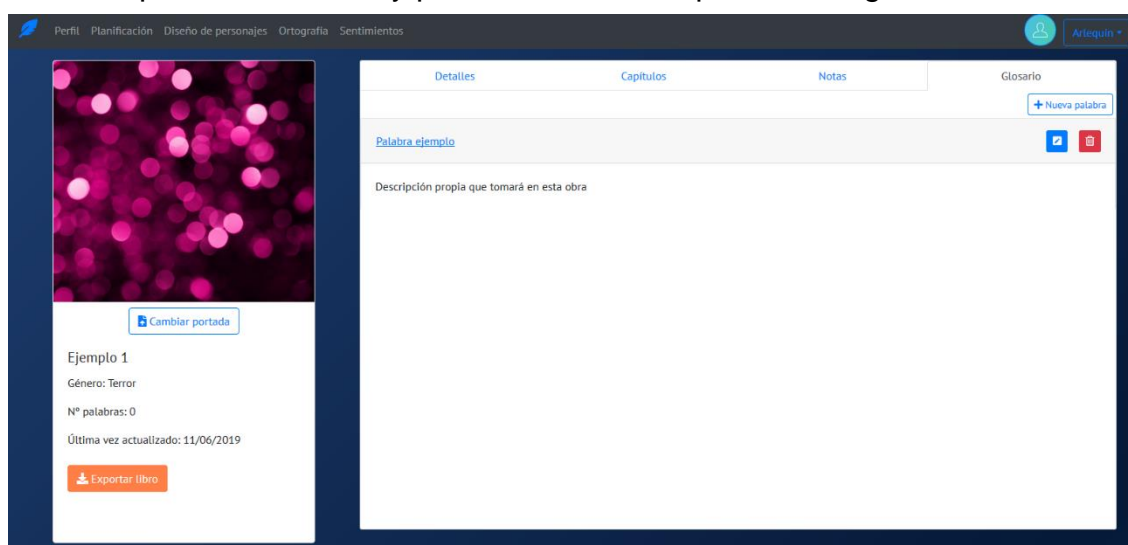
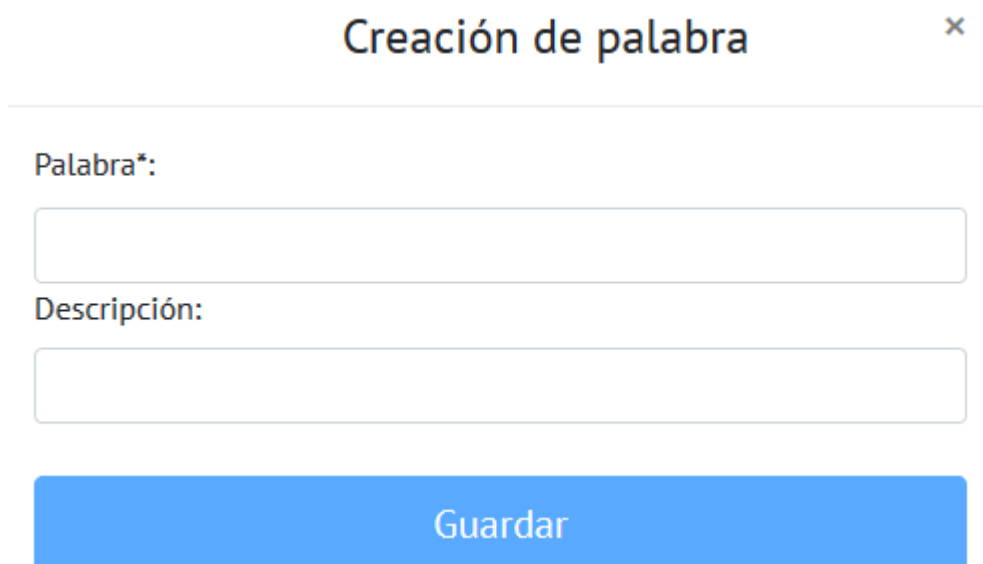


Figura 34 – Página del glosario asociado al libro

De esta forma aparece el listado de palabras en el glosario en forma de acordeón al igual que el diccionario de palabras.

El escritor será el encargado de alimentar la lista de palabras al igual que en el propio diccionario, por lo tanto, el formulario de creación y edición de palabras es similar al del diccionario de palabras.



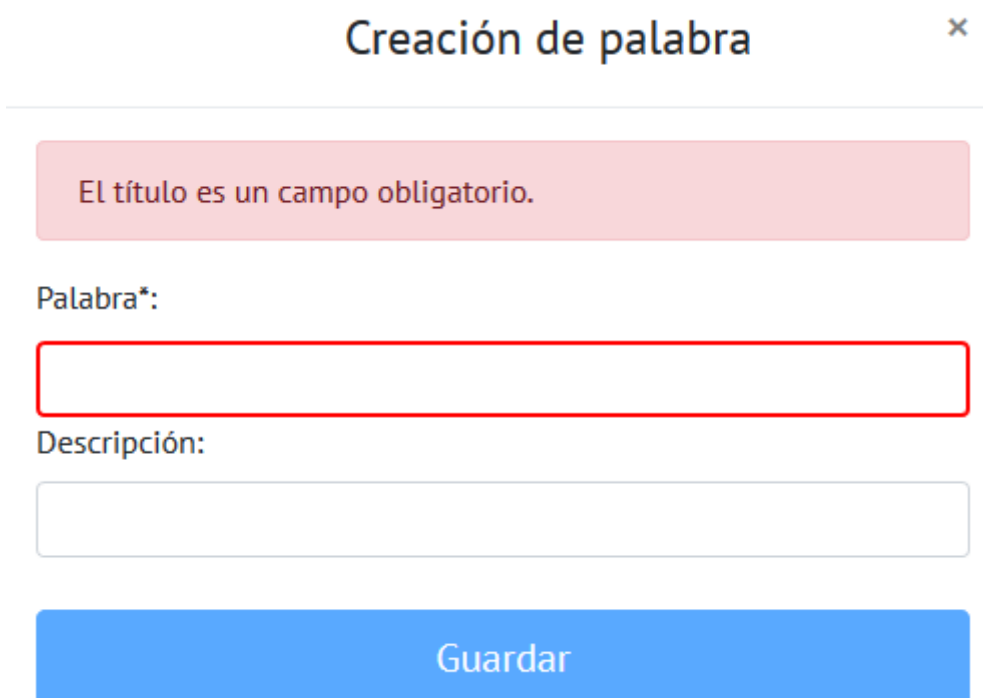
Creación de palabra ×

Palabra*:

Descripción:

Guardar

Figura 35 – Formulario de creación de palabra del glosario



Creación de palabra ×

El título es un campo obligatorio.

Palabra*:

Descripción:

Guardar

Figura 36 – Formulario de creación de palabra del glosario erróneo mostrando advertencia

Edición de palabra ×

Palabra*:

Palabra ejemplo

Descripción:

Descripción propia que tomará en esta obra

Guardar

Figura 37 – Formulario de modificación de palabra del glosario

El escritor deberá de rellenar el formulario para crear/modificar la palabra del glosario del libro seleccionado. Este formulario como campo obligatorio solo tiene el campo palabra, que si no está informado aparecerá un mensaje de error para informar al escritor.

En el caso de que el escritor desees eliminar alguna palabra del glosario aparecerá una ventana de confirmación.

Eliminar palabra ×

¿Estás seguro de que quieres borrar Palabra ejemplo?

Toda la información relacionada con esta palabra será permanentemente borrada. Esta operación no se puede deshacer

Cancelar Eliminar

Figura 38 – Página de confirmación de borrado de palabra del glosario

Cada libro tiene un listado de notas asociado

El escritor por cada libro que crea tiene acceso a un listado de notas propio de cada libro en el que añadir notas. Estas notas que se definen en el listado de cada libro son creadas por el escritor por que quiera aclarar hechos que ocurran en el libro, ampliar alguna de las explicaciones que se dan en el libro a modo de extender la obra y hacerlo en la propia trama podría ser una distracción. El listado de notas se encuentra en una de las pestañas en la sección del libro seleccionado, la pestaña en cuestión es *notas*.

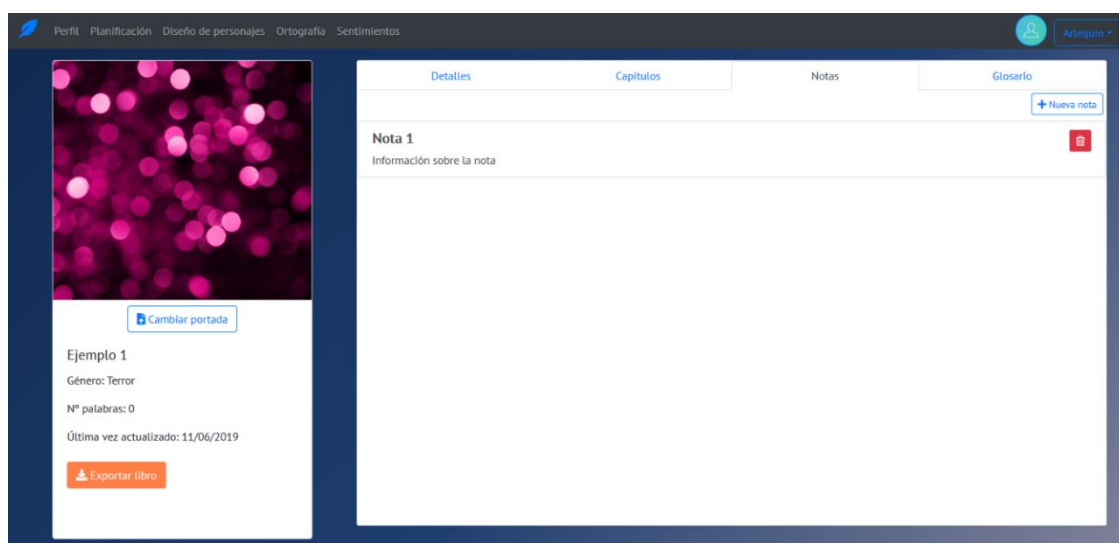


Figura 39 – Página de notas asociadas al libro

El listado de notas de cada libro es alimentado por el propio escritor por lo que podrá añadirlas pulsando el botón para añadir una nueva nota y rellenando el formulario en cuestión.

Creación de una nota

×

Título*:

Descripción:

Guardar

Figura 40 – Formulario de creación de notas

Creación de una nota

×

El título es un campo obligatorio.

Título*:

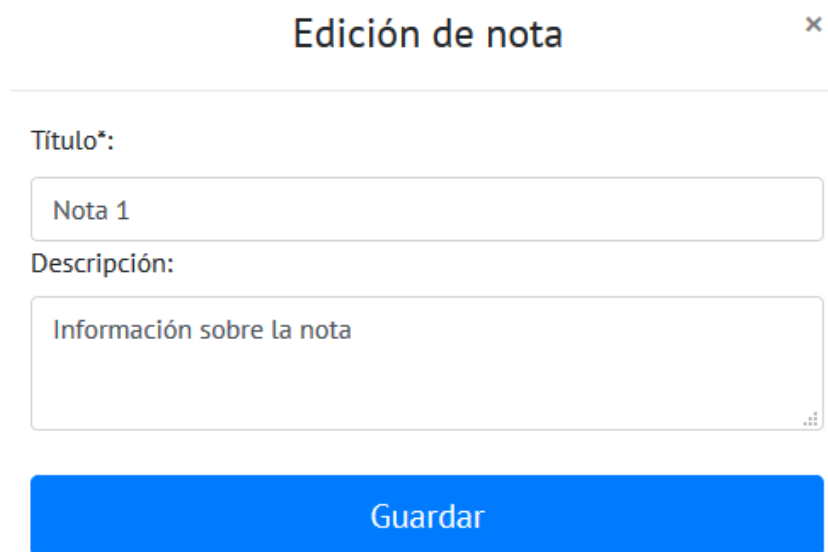
Descripción:

Guardar

Figura 41 – Formulario de creación de notas erróneo con advertencia

El formulario de creación de una nueva tarea está compuesto por dos campos que debe informar el escritor, en primera instancia será el título de la nota, el cual es obligatorio, y en segunda instancia una descripción para extender lo que el escritor desea explicar o esclarecer. Después el escritor podrá ver la nueva nota añadida en el listado.

En el caso de que el escritor quiera modificar alguna nota de las creadas solo debe de pinchar sobre la nota en cuestión y aparecerá la información de esa nota para ser modificada.



Edición de nota x

Título*:

Nota 1

Descripción:

Información sobre la nota

Guardar

Figura 42 – Formulario de modificación de notas

El último caso es que el escritor quiera eliminar alguna de las notas creadas por cualquier cuestión. Solo deberá de pinchar el botón en cuestión y aparecerá la ventana de confirmación para ser eliminado.

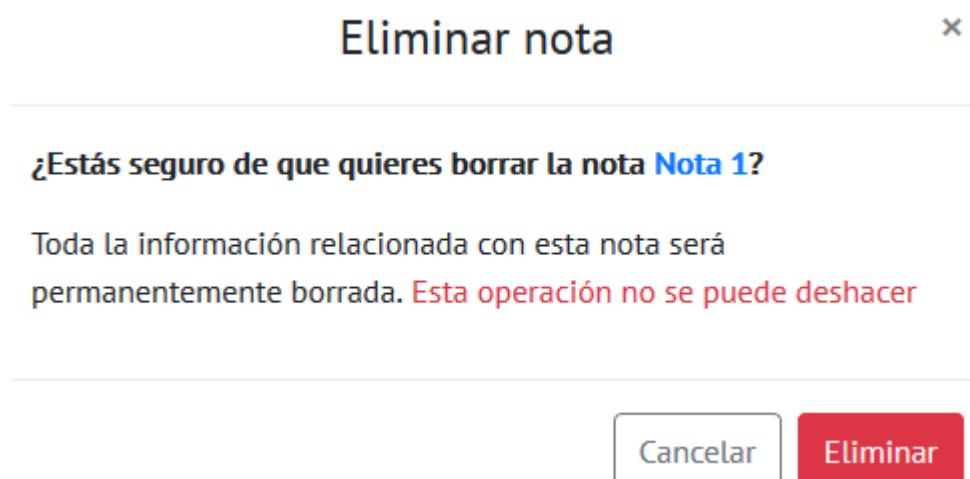


Figura 43 – Página de confirmación de borrado de una nota

El escritor tendrá una galería de personajes

Uno de los accesos directos que el escritor encuentra en la barra superior de navegación es el *diseño de personajes*. Al acceder a esta sección el usuario se encontrará con todos los personajes que él ha creado en el transcurso del diseño de sus obras. Cuando un escritor decide escribir necesita de personajes que vivan y participen en la historia que va a contar, por lo tanto, necesita definir de manera muy clara y convincente para que dichos personajes sean creíbles por el futuro lector y consiga atraparlos completamente.

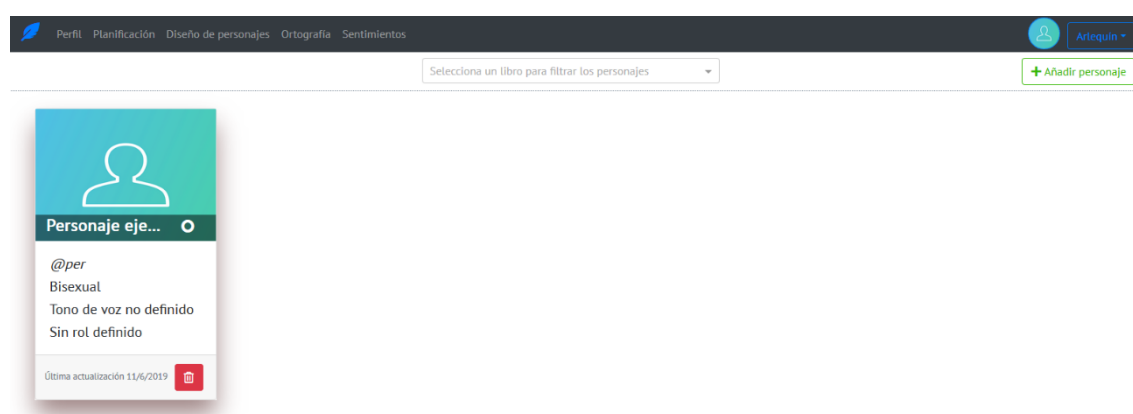


Figura 44 – Página de la galería de personajes

Quando el escritor entra en la sección de diseño de personajes verá una galería de todos los personajes sin filtrar, si tiene alguno ya creado. En el caso

de que tenga varios libros y para cada libro tenga varios personajes creados al principio saldrán todos mezclados, pero el escritor tiene en la parte superior central un elemento para elegir un libro, y por lo tanto solo aparecerán los personajes asociados a dicho libro seleccionado.

Para añadir nuevos personajes el usuario solo tiene que pinchar sobre el botón y lo redirigirá al formulario de creación.

The screenshot shows the 'Creación de un nuevo personaje' form with the 'Esencial' tab selected. The form includes a header with navigation links (Perfil, Planificación, Diseño de personajes, Ortografía, Sentimientos) and a user profile icon. The main content area is divided into three sections: 'Esencial', 'Aspecto', and 'Personalidad'. The 'Esencial' section contains a large blue square with a white person icon and a 'Cambiar imagen' button. Below this, there is a list of status messages: 'Sin alias aún', 'Falta definir su orientación', 'Tono de voz no definido', 'Sin rol definido', and 'Última actualización'. The 'Aspecto' section contains several input fields and dropdown menus for character details: 'Nombre*' (Nombre del personaje), 'Alias' (Alias sobre el personaje), 'Rango de edad' (0 - 0 to 100), 'Género' (Genero del personaje), 'Lugar de nacimiento' (¿Dónde ha nacido?), 'Orientación sexual' (Orientación sexual), 'Rol' (Rol del personaje), 'Ancestros' (Ancestros del personaje), 'Hermanos' (Hermanos del personaje), 'Hijos' (Hijos del personaje), and 'Parejas' (Pareja/s del personaje). The 'Personalidad' section contains a 'Biografía' text area with the prompt 'Describe más sobre el personaje...'. At the top right, there is a 'Guardar' button and a 'Selecciona el libro al que pertenece' dropdown menu.

Figura 45 – Formulario esencial del diseño de personajes

The screenshot shows the 'Creación de un nuevo personaje' form with the 'Aspecto' tab selected. The form includes a header with navigation links (Perfil, Planificación, Diseño de personajes, Ortografía, Sentimientos) and a user profile icon. The main content area is divided into three sections: 'Esencial', 'Aspecto', and 'Personalidad'. The 'Aspecto' section contains several input fields and dropdown menus for character details: 'Compleción' (Compleción del personaje), 'Tipo de cuerpo' (Tipo de cuerpo), 'Altura en centímetros (100cm = 1m)' (0 to 300), 'Vestimenta' (Tipo de vestimenta), 'Tono de voz' (Tono de voz), 'Volumen de voz' (Volumen de voz por defecto), 'Tono de piel' (Tono de la piel), 'Tipo de piel' (Tipo de piel), 'Gesticulación' (Tipo de gesticulación), 'Vello corporal' (Tipo de vello), 'Color de pelo' (Color del pelo), 'Tamaño del pelo' (Tamaño del pelo), 'Color de ojos' (Color de los ojos), 'Tipo de ojos' (Tamaño de los ojos), 'Tipo de nariz' (Tipo de nariz), and 'Tipo de orejas' (Tipo de orejas). At the top right, there is a 'Guardar' button and a 'Selecciona el libro al que pertenece' dropdown menu.

Figura 46 – Formulario del aspecto del diseño de personajes

Perfil Planificación Diseño de personajes Ortografía Sentimientos

Creación de un nuevo personaje

Selecciona el libro al que pertenece Elige un libro + Guardar

Esencial Aspecto Personalidad

Motivaciones/Objetivos
Describe las motivaciones y objetivos que tiene el personaje.

Manías
Describe manías, rarezas y formas únicas que tenga el personaje.

Aficiones
Describe aficiones, gustos y pasiones que tenga el personaje.

Forma de comportarse
Describe el comportamiento del personaje en estado natural, ante alguna amenaza, bajo presión...

Creencias/Valores
Describe las creencias y valores que forjan la personalidad del personaje.

Miedos
Describe los miedos y temores que tiene.

Figura 47 – Formulario de personalidad del diseño de personajes

Este formulario está dividido en 3 partes, el cual intenta abarcar los aspectos esenciales para dar forma a los personajes que van a pertenecer a las obras. De todos los campos pedidos en la creación de personajes solo hay dos campos necesarios rellenarlos obligatoriamente, el nombre y el libro al que pertenece, si no se rellena alguno de estos dos campos aparecerá una advertencia para avisar al escritor. Una vez rellenados los campos necesarios solo deberá apretar el botón de guardar y el nuevo personaje aparecerá en la galería con los demás personajes.

Si el escritor desea modificar cualquier dato de algún personaje solo debe seleccionarlo y todos sus datos se cargarán en el formulario pudiendo modificar y guardarlo de nuevo.

Para la acción de borrar alguno de los personajes borrado aparecerá una pantalla de confirmación, la cual tendrá que aceptar el escritor para que se lleve a cabo la eliminación.

Eliminar personaje



¿Estás seguro de que quieres borrar al personaje **Personaje ejemplo?**

Toda la información relacionada con esta tarea será permanentemente borrada. Esta operación no se puede deshacer

Cancelar

Eliminar

Figura 48 – Formulario de confirmación para el borrado de algún personaje

Toda la información que rellena el escritor mediante el formulario de creación/modificación es enviado al controlador encargado de almacenarlo en la base de datos. Dicho controlador mapeará el objeto de character que le llega por una de las interfaces definidas y mapeará con el modelo de datos definidos en el back-end, el cual abrirá la conexión con la base de datos para almacenar en una tabla propia llamada Characters. Como se comentó en el modelo de datos la gestión de los identificadores se hace de forma manual, y para saber de donde es cada personaje se identifica a través del WriterId el cual indica al escritor al que pertenece y el BookId que indica a que libro pertenece.

El escritor podrá acceder a un planificador de tareas

Uno de los accesos directos que el escritor encuentra en la barra superior de navegación es la *planificación*. Al acceder a esta sección el usuario se encontrará una pantalla con tres columnas que indican lo siguiente:

- 1- Tareas pendientes
- 2- Tareas en curso
- 3- Tareas completadas

Estas 3 columnas permitirán al escritor organizarse clasificando las tareas que tiene pendientes y aun no se ha puesto a realizarlas, las tareas en curso o activas las cuales indican que el escritor está trabajando en ellas y las tareas ya completadas que indican que esas tareas ya están terminadas.

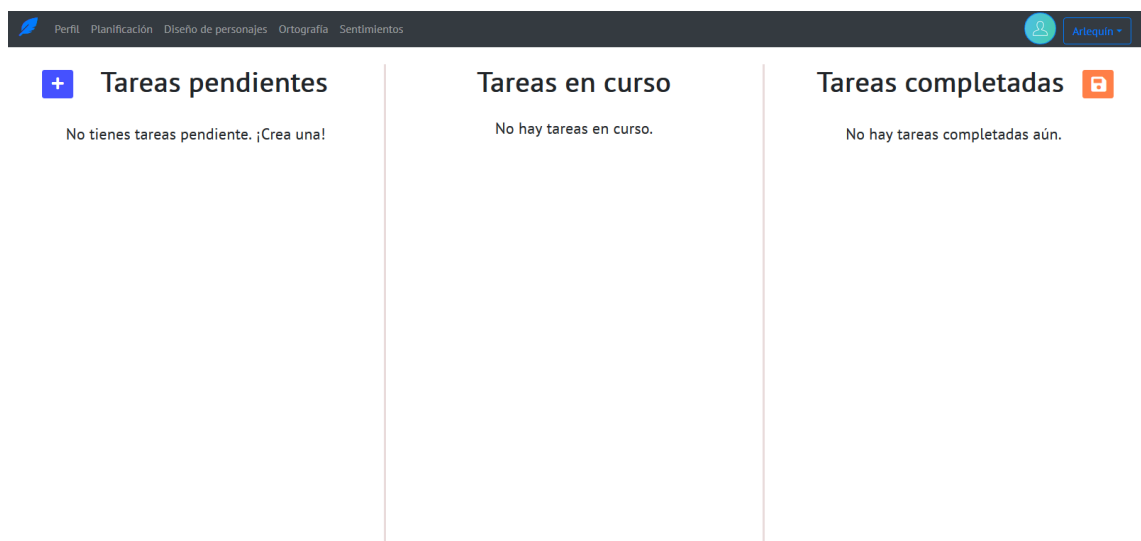


Figura 49 – Pagina del planificador de tareas

El escritor podrá añadir nuevas tareas a la lista de tareas pendientes pulsando el botón en la columna de *tareas pendientes*. Al pulsar para añadir una nueva tarea se abrirá el formulario el cual deberá rellenar el escritor.

Creación de tarea

Título*:

Fecha límite

dd/MM/yyyy

Descripción:

Crear

Figura 50 – Formulario de creación de una tarea

En este formulario el escritor especificará un título identificativo que es obligatorio junto con la fecha límite de realización de la tarea y, una descripción por si debe añadir aclaraciones acerca de la tarea. Si no se rellenan los campos obligatorios aparecerán unos mensajes informativos para que el escritor esté informado.

Creación de tarea

El título es un campo obligatorio.

La fecha límite es un campo necesario para organizarte.

Título*:

Fecha límite

dd/MM/yyyy

Descripción:

Figura 51 – Formulario de creación de una tarea erróneo con advertencias

Al pulsar crear aparecerá la tarea en la lista de tareas pendientes. Una funcionalidad muy interesante es que las tareas de las tres columnas se pueden cambiar de columna arrastrando desde una columna hasta la columna destino, por lo tanto, es muy dinámica la gestión y planificación de tareas. Una vez que se han acabado de mover las tareas entre las columnas deberá pulsar el botón

de guardado situado en la columna de *tareas completadas*, y aparecerá una confirmación para el guardado.

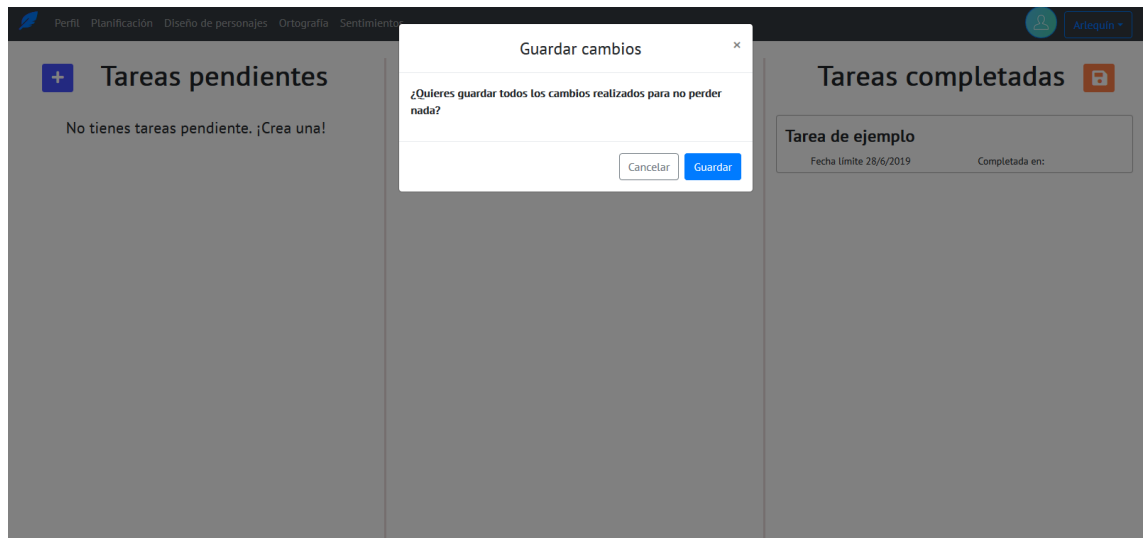



Figura 52 – Página de confirmación de guardado de tareas

Mientras las tareas permanecen en las columnas de tareas pendientes o tareas en curso aparecerá un botón en la propia tarea para poder modificar la

 **Modificación de tarea** ×

Identificador*:

Título*:

Descripción:

Alguna aclaración.

Completar

Fecha límite 28/6/2019

Guardar

información de ella misma. Al pulsar dicho botón se abrirá un formulario para editar cierta información.

Figura 53 – Formulario de modificación de tarea

En este formulario el escritor puede modificar el título y la descripción, además puede eliminar dicha tarea seleccionada la cual pedirá una confirmación como siempre se pide en cualquier eliminación de algún elemento en toda la plataforma. También puede completar la tarea pulsando el botón de *Completar* el cual hace que la tarea pase a tener el estado de completada y asignarle la fecha de cierre.

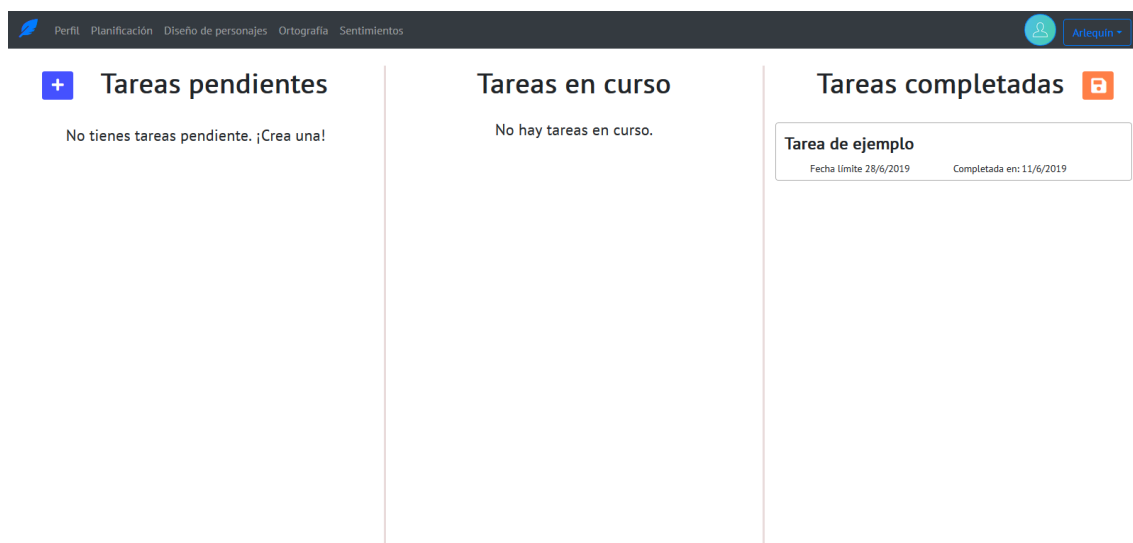


Figura 54 – Página de tareas con una tarea completada

La lista de tareas como se ha visto en el modelo de datos está incrustada en el modelo del escritor por lo tanto al ser una lista hay un campo que indica el estado en el que se encuentra la tarea y por lo tanto al recibirla en el front-end se divide en tres listas, una para cada columna. Esta gestión de listas finalmente si se modifica cualquier tarea o se añaden nuevas se tiene que convertir en una única lista que será la que se enviará al controlador para que lo guarde la nueva lista. Al estar incrustado en el modelo del escritor nunca se hacen peticiones de creación a la Api como son las peticiones Post, solo se envían peticiones de actualización Put ya que en realidad se está modificando es un campo del modelo del Writer.

El escritor podrá tener un análisis de sentimientos de sus obras

El escritor cuando está escribiendo puede no pararse a revisar los capítulos escritos y tener una concepción de lo que ha escrito un poco diferente a como lo pensó en un primer momento, por lo tanto, el escritor podrá acceder a través de los enlaces directos de la barra superior de navegación pinchando en *sentimientos*.

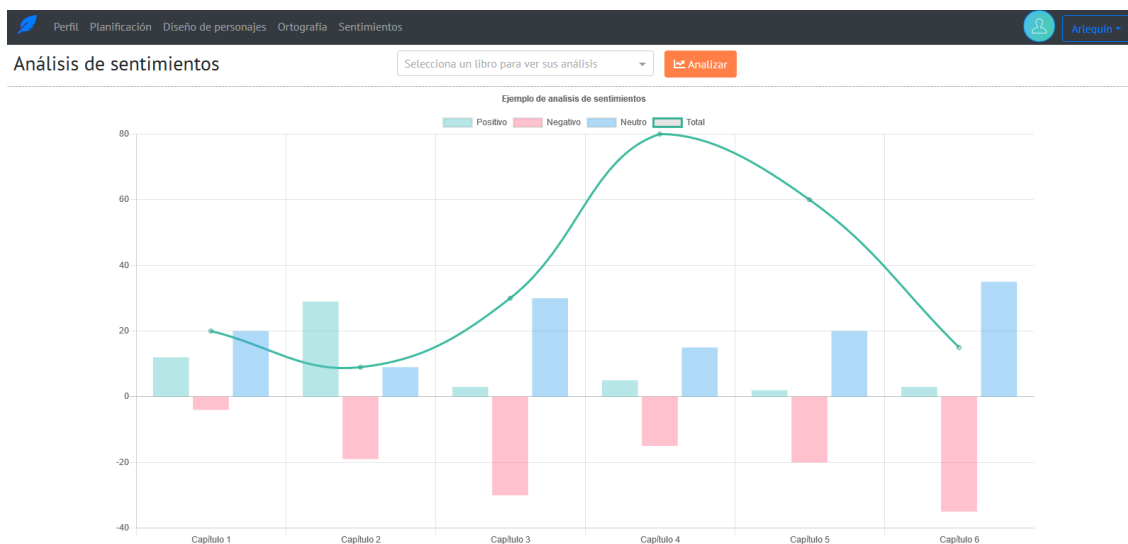


Figura 55 – Página análisis de sentimientos con gráfica de ejemplo

En esta sección el usuario podrá ver de forma muy visual el análisis de sentimientos que puede realizarle a las obras que esté creando. Dicho análisis lo puede realizar eligiendo un libro, en la lista desplegable central superior, y pulsando el botón de analizar. Sí la obra no contiene ningún análisis previo no aparecerá ninguna función ni diagrama en la gráfica e indicará al escritor que no tiene ningún análisis en la parte de la leyenda de la gráfica.

Si se selecciona un libro y se decide analizar o es un libro con un análisis previo ya realizado, el escritor podrá observar cómo los valores se cargan en forma de dos modelos de gráficas. Como se puede ver en el ejemplo intervienen dos modelos de gráficas, una función lineal y un diagrama de barras.

En el eje de abscisas se encuentran los capítulos que el libro seleccionado tiene creados, y en el eje de ordenadas se indica desde 0 a 100, donde 0 indica que ese capítulo no transmite ningún sentimiento y 100 un sentimiento muy positivo, esto lo representa la función lineal. Para el diagrama de barras se puede ver que hay tres barras:

- La barra de color roja se representará de forma negativa para una mayor claridad y en ella se muestran el número de palabras o segmentos negativos.
- La barra de color azul representa las palabras que transmiten un sentimiento neutral o no transmiten ningún tipo de sentimiento, en este caso se representa de forma positiva y refleja el número de palabras o segmentos neutrales.
- Y por último la barra de color verde representa las palabras que transmiten un sentimiento positivo y refleja el número de palabras o segmentos positivos en el texto analizado.

En cada capítulo se puede observar un punto de la función lineal y tres barras las cuales reflejan el análisis realizado al libro dividido en capítulos.

Para la realización del análisis de sentimientos de las obras escritas se está utilizando una Api externa la cual se envía el texto a analizar y devuelve una respuesta en formato Json, el cual se analizará centrándose en los elementos principales como son *score_tag*, que refleja el sentimiento de la palabra o segmento detectado. El resultado que devuelve la propiedad *score_tag* no está contenida entre un número de cero a cien, sino que devuelve una nota especial categorizada y clasificada como:

- muy positiva(N+)
- positiva(N)
- neutral (NEU)
- sin sentimiento ()
- negativa(N)
- muy negativa(N+)

Por lo tanto, se ha decidido transformar dicha nota especial en una nota numérica para poder representarla con la función lineal, donde cien es la nota máxima positiva, dividida entre 6, número de notas devueltas por dicha Api. La nota numérica mínima es asignada a la nota de no transmitir ningún sentimiento, devuelta por la Api.

Para ver el modelo de respuesta que devuelve la Api y como se ha procedido a recabar la información es necesario un ejemplo. El modelo de datos devuelto es extenso y complejo, tanto como se detecten estructuras anidadas y palabras que transmiten algún tipo de sentimiento.

Para el ejemplo vamos a usar parte de un texto de *Guillem Menelik Parra*:

“Al ver Etiopía con mis propios ojos, me sentí orgulloso por mi sangre, sangre abisinia, procedente del único país que nunca ha sido colonizado; un orgullo que empezó con el reino de Axum y sigue en todos los etíopes del mundo.” (Parra, 2019)

La respuesta que se obtendría es:

```
{
  status: {
    code: "0",
    msg: "OK",
    credits: "1",
    remaining_credits: "19992"
  },
  model: "general_es",
  score_tag: "P",
  agreement: "AGREEMENT",
  subjectivity: "SUBJECTIVE",
  confidence: "100",
  irony: "NONIRONIC",
  sentence_list: [
    { 10 items },
    { 10 items }
  ],
  sentimented_entity_list: [ 2 items ],
  sentimented_concept_list: [ 5 items ]
}
```

Como se puede ver la respuesta se han colapsado algunos elementos para poder ver la información de primer nivel, además, las propiedades que se pueden ver hacen referencia al sentimiento que transmite el texto de forma global, por lo tanto, el score_tag que se observa hace referencia a la puntuación de la función lineal.

Además del sentimiento que transmite el texto, se recibe información de si el texto enviado

Figura 56 – Respuesta API sentimientos

contiene ironía y si es objetivo o subjetivo. Para formar el diagrama de las tres barras el algoritmo se focaliza en las listas sentence_list, sentimented_entity_list y sentimented_concept_list.

```

sentence_list: [
  {
    text: "Al ver Etiopía con mis propios ojos,
me sentí orgulloso por mi sangre,
sangre abisinia, procedente del único
país que nunca ha sido colonizado;",
    inip: "0",
    endp: "142",
    bop: "y",
    confidence: "100",
    score_tag: "p",
    agreement: "AGREEMENT",
    segment_list: [
      {
        text: "Al ver Etiopía con mis propios ojos,
me sentí orgulloso por mi sangre,
sangre abisinia, procedente del único
país que nunca ha sido colonizado",
        segment_type: "main",
        inip: "0",
        endp: "141",
        confidence: "100",
        score_tag: "p",
        agreement: "AGREEMENT",
        polarity_term_list: [...],
        sentimented_entity_list: [...],
        sentimented_concept_list: [...]
      }
    ]
  }
],

```

Si se despliega uno de los nodos del `sentence_list` se puede ver información sobre frases completas. Además, esta frase puede tener múltiples segmentos y, por lo tanto, el algoritmo para la búsqueda de todas las notas ha sido recursivo, para poder hacer una búsqueda en profundidad en la jerarquía que se puede montar en el texto analizado. Al acabar de buscar en la lista de segmentos, `segment_list`, que es la que presenta la estructura de recursividad, el algoritmo

Figura 57 – Respuesta API

sentimientos segmentos

seguiría con las palabras de las otras listas. Finalmente se obtendría un número de palabras positivas, negativas y neutras las cuales se representarían en forma de diagramas de barras, como se ha mencionado con anterioridad.

Ese análisis ayuda al escritor a saber cómo son sus capítulos y que sentimientos están transmitiendo, pudiendo así corregir alguno de ellos si el fin que él quiere no coincide con el resultado del análisis.

Todo este análisis que devuelve la Api es almacenado en la base de datos en una colección propia, por lo tanto, se realizan operaciones de Post, Put, Get y Delete para interaccionar con la base de datos.

Gracias a la integración con esta Api externa ha permitido realizar unos análisis de sentimientos en los textos que el escritor puede crear y tener una retroalimentación de lo que transmite su obra. Está Api es parte de la empresa MeaningCloud, la cual está especializada en análisis semántico. Y se ha podido comprobar que el resultado que proporcionan sus Apis es muy positivo.

7

Pruebas de la aplicación

Pruebas unitarias

La parte de pruebas de un desarrollo de software es una etapa más dentro de la planificación de cualquier proyecto, pero vital y necesaria para construir un proyecto de calidad. El objetivo principal de dichas pruebas es verificar que el desarrollo cumple con unos objetivos preestablecidos, los cuales indican si el software construido posee la calidad esperada o no.

En esta etapa se pueden encontrar una gran variedad de pruebas que se pueden aplicar a los desarrollos, y cada uno de estos tipos de pruebas consiguen comprobar cierto aspecto de la aplicación.

Estas pruebas pueden ir desde test unitarios, hasta test del sistema o test de estrés entre otros. Cada tipo de pruebas dan cierta información, lo más objetiva posible, para poder verificar que los objetivos se están cumpliendo y además ver si el código escrito es de calidad, si tiene malas prácticas, grado de mantenibilidad de la aplicación, complejidad del código, etc. Todas estas puntuaciones hacen que se tomen decisiones para corregir aquellas en las que se obtenga peores puntuaciones.

Para la etapa de pruebas de este proyecto se ha decidido enfocarse sobre las pruebas unitarias de la Api, que es la encargada de conectarse a la base de datos y enviar los datos correspondientes a la parte del front-end. Se ha acordado realizar solo estas pruebas porque eran las que se ajustaban mejor al tiempo que tenía el proyecto planificado.

La guía principal que se ha seguido para la realización de cada una de las pruebas ha sido el principio de las tres A's. Este principio hace que cada prueba se divida a su vez en tres partes diferenciadas:

- Arrange (Organización): Esta etapa sirve para hacer toda la preparación de los elementos del sistema que van a intervenir y ser probados en la prueba a realizar.
- Act (Actuación): Etapa donde se realizan las llamadas a los métodos correspondientes.
- Assert (Comprobación): Etapa donde se verifican y comprueban que los resultados a las llamadas de la etapa anterior son correctos.

Estas pruebas unitarias han sido de gran utilidad a la hora de mantener la Api con un código más limpio y de calidad, eliminando duplicidad de código y

métodos sin utilidad, haciendo que dichos controladores sean más manejables, escalables y más sencillos de mantener. Para cada acción de los controladores que se usa en la Api, Post, Put, Get y Delete, se han realizado pruebas unitarias cubriendo todos los caminos que podía tomar el flujo de control.



Figura 58 – Pruebas correctas de los controladores

Como se puede observar se han construido cuarenta pruebas unitarias, que hacen que los controladores involucrados en la parte del proyecto que aquí se desarrolla, tenga una calidad y seguridad de que cumple con los requisitos establecidos en los objetivos. Para la realización de las pruebas se ha tenido que poner especial atención en los elementos que se querían probar, ya que puede llevar a una fácil confusión e intentar hacer test unitarios a cajas negras de terceros, que no son de la incumbencia de este proyecto.

Hierarchy	Not Covered (Blocks)	Not Covered (% Blocks)	Covered (Blocks)	Covered (% Blocks)
jesus_DESKTOP-6KO2074 2019-06-19 23_50_19.coverage	2276	57,94 %	1652	42,06 %
correctorapp.web.dll	1830	81,05 %	428	18,95 %
CorrectorApp.Web.Data.MongoModels.Writers	392	97,03 %	12	2,97 %
CorrectorApp.Web.Controllers	308	45,63 %	367	54,37 %
CorrectorApp.Web.Data.MongoModels.Sentimen...	182	100,00 %	0	0,00 %
CorrectorApp.Web	179	100,00 %	0	0,00 %
CorrectorApp.Web.Data.MongoModels.TextAnaly...	176	96,70 %	6	3,30 %
CorrectorApp.Web.Data.MongoModels.Books	138	97,87 %	3	2,13 %
CorrectorApp.Web.Data.MongoModels.Sentimen...	104	100,00 %	0	0,00 %
CorrectorApp.Web.CognitiveServices	83	100,00 %	0	0,00 %
CorrectorApp.Web.Data.MongoModels.Characters	74	90,24 %	8	9,76 %
CorrectorApp.Web.Controllers.Utils	64	79,01 %	17	20,99 %
CorrectorApp.Web.Data.MongoModels	57	100,00 %	0	0,00 %
Global Classes	50	100,00 %	0	0,00 %
CorrectorApp.Web.Pages	14	100,00 %	0	0,00 %
CorrectorApp.Web.Data.Models	6	42,86 %	8	57,14 %
CorrectorApp.Web.Data.Models.Response	3	30,00 %	7	70,00 %
testcorrectorapp.web.dll	446	26,71 %	1224	73,29 %

Figura 59 – Estadísticas de análisis de boques de código

Todas estas métricas de cobertura de líneas y bloques están basadas en las pruebas unitarias a cada controlador, y por lo tanto dan una información muy valiosa y visual a simple vista donde se puede ver en qué partes del proyecto se aplican las pruebas y qué componentes son las que validan dichas pruebas.

Hierarchy	Not Covered (Lines)	Not Covered (% Lines)	Covered (Lines)	Covered (% Lines)
jesus_DESKTOP-6KQ2074 2019-06-19 23_50_19.coverage	1551	61,62 %	964	38,30 %
testcorrectorapp.web.dll	214	27,02 %	578	72,98 %
correctorapp.web.dll	1337	77,51 %	386	22,38 %
{ } CorrectorApp.Web.Data.Models.Response	3	30,00 %	7	70,00 %
{ } CorrectorApp.Web.Data.Models	6	42,86 %	8	57,14 %
{ } CorrectorApp.Web.Pages	6	100,00 %	0	0,00 %
{ } Global Classes	50	100,00 %	0	0,00 %
{ } CorrectorApp.Web.Data.MongoModels	59	100,00 %	0	0,00 %
{ } CorrectorApp.Web.Controllers.Utils	54	65,06 %	29	34,94 %
{ } CorrectorApp.Web.Data.MongoModels.Characters	74	90,24 %	8	9,76 %
{ } CorrectorApp.Web.CognitiveServices	64	100,00 %	0	0,00 %
{ } CorrectorApp.Web.Data.MongoModels.Sentimen...	104	100,00 %	0	0,00 %
{ } CorrectorApp.Web.Data.MongoModels.Books	87	96,67 %	3	3,33 %
{ } CorrectorApp.Web.Data.MongoModels.TextAnaly...	90	93,75 %	6	6,25 %
{ } CorrectorApp.Web	201	100,00 %	0	0,00 %
{ } CorrectorApp.Web.Data.MongoModels.Sentimen...	99	100,00 %	0	0,00 %
{ } CorrectorApp.Web.Controllers	245	43,75 %	313	55,89 %
{ } CorrectorApp.Web.Data.MongoModels.Writers	195	94,20 %	12	5,80 %

Figura 60 – Estadísticas de análisis de líneas de código

Además, Visual Studio 2017 ofrece algunas herramientas integradas que hacen este proceso muy cómodo y sencillo, presentando de forma muy visual los bloques/líneas de código por donde las pruebas unitarias han ejecutado las pruebas, en azul, y qué líneas de código aún no han sido cubiertas por ninguna prueba, en rojo.

Para la realización de estas pruebas unitarias se ha utilizado un paquete Nuget llamado Xunit, el cual permite realizar pruebas como las comentadas en entornos .Net Core. Este paquete junto con Visual Studio 2017 permite desarrollar pruebas unitarias a una gran velocidad permitiendo verificar y encontrar errores en el código que de otra forma sería imposible detectar.

```

// POST api/writers/newuser - creates a new writer
[HttpPost]
[Route("newuser")]
public IActionResult Post([FromBody] Writer newWriter)
{
    try
    {
        if (_writerRepository.SearchByEmail(newWriter.Email))
        {
            return Ok(new HttpResponseMessage<Writer>()
            {
                Code = 200,
                Status = StatusHttp.Response.Error,
                IsFaulted = false,
                Message = $"Usuario con email {newWriter.Email} ya existe en el sistema. Intenta iniciar sesión.",
                ListObjects = null
            });
        }
        else
        {
            newWriter.Password = AlgoEncrypt.Encrypt(newWriter.Password, out string salt);
            newWriter.Salt = salt;
            newWriter.CreatedOn = DateTime.Now.ToString("dd MMM yyyy");
            newWriter.WriterId = this.GetCountDocuments().Result.ToString();
            var response = _writerRepository.AddWriter(newWriter);

            return Ok(new HttpResponseMessage<Writer>()
            {
                Code = 201,
                Status = StatusHttp.Response.Created,
                IsFaulted = false,
                Message = $"Usuario con email {newWriter.Email} creado correctamente. Ya puedes iniciar sesión y empezar la aventura.",
                ListObjects = null
            });
        }
    }
    catch (Exception e)
    {
        return BadRequest(new HttpResponseMessage<Writer>()
        {
            Code = 400,
            Status = StatusHttp.Response.Error,
            IsFaulted = true,
            Message = e.Message,
            ListObjects = null
        });
    }
}

```

Figura 61 – Líneas de código cubierta por las pruebas unitarios

Test de usabilidad

Otras de las muchas pruebas que se le pueden realizar a un sistema, en este caso a una aplicación web, es que usuarios reales prueben e interactúen con la propia aplicación haciendo una valoración contestando una serie de preguntas.

Se ha tomado como referencia el sistema de escala de usabilidad (System usability scale) por ser un estándar simple y exacto de diez preguntas, con un sistema de respuestas comprendidas del 0 al 5, donde 0 es «completamente en desacuerdo» y 5 es «completamente de acuerdo». Con estas valoraciones de los usuarios se obtiene una retroalimentación muy positiva, ya que en una etapa temprana se puede saber si hay que replantear y rediseñar alguna sección de la aplicación.

Para hacer estas pruebas se han dividido la muestra en tres usuarios:

- Usuario inexperto
- Usuario medio

- Usuario experto

A continuación, se muestra la tabla que recoge las preguntas y correspondientes respuestas de cada usuario.

Preguntas	Usuario inexperto	Usuario medio	Usuario experto
Creo que me gustaría utilizar este sistema frecuentemente.	3	5	5
El sistema me resultó innecesariamente complejo.	3	3	3
Creo que el sistema es bastante fácil de utilizar.	2	3	4
Creo que necesitaría el soporte de un técnico para poder utilizar este sistema.	5	5	2
Creo que las diferentes funciones del sistema se encuentran muy bien integradas.	4	5	5
Opino que hubo demasiada inconsistencia en el sistema.	1	1	1
Imagino que la mayoría de las personas aprendería a utilizar el sistema rápidamente.	2	4	4
Me sentí algo incómodo al utilizar este sistema.	2	1	1
Me sentí muy seguro al utilizar este sistema.	5	5	5
Necesito aprender muchas otras cosas antes de poder utilizar correctamente el sistema.	5	5	2

Tabla 52 – Preguntas de usabilidad para usuarios

8

Posibles extensiones

Al desarrollar este proyecto se han ido ocurriendo muchas ideas las cuales han quedado fuera de la planificación y se han priorizado los requisitos que pueden aportar más valor al usuario final, los escritores.

Empezando desde lo más sencillo se podría añadir un sistema de recuperación de contraseña por si el escritor ha olvidado con qué contraseña se registró. Permitir más formas de registro en la plataforma como puede ser a través de la cuenta de Google, Facebook, LinkedIn, GitHub entre otras tantas posibilidades..., introducir todas estas formas de identificación en el sistema conllevaría en adaptar y mejorar la gestión de los tokens de acceso.

Entrando en el tema del diseño de interfaces de usuario, siempre se puede mejorar, por lo tanto, se podría extender haciendo que la plataforma sea una aplicación totalmente adaptable a cualquier dispositivo, desde un teléfono móvil hasta pantallas grandes.

Al estar usando la versión cinco de Angular se ha tenido que tomar la decisión de acoplar componentes externos que otorgasen una funcionalidad específica, por lo tanto, como Google está sacando actualizaciones con bastante asiduidad, podría ser beneficioso actualizar a las versiones más actuales de las tecnologías usadas para el desarrollo de este proyecto. Hasta el momento se ha anunciado la versión ocho de Angular, actualizar a dicha versión aportaría eliminar algunos componentes externos y adoptar dicha funcionalidad de la manera nativa que ofrecen las nuevas versiones de Angular. Un caso claro es en el planificador de tareas, donde la acción de poder mover tareas entre varias columnas está proporcionada por una librería externa y en el caso de Angular siete ya está de forma nativa, dando un mayor control al desarrollador y eliminando una dependencia de esa librería externa.

Otras mejoras que podrían tener cabida en este proyecto serían aquellas que proporcionen al escritor una herramienta para el desarrollo de la obra como puede ser un buscador de palabras, sinónimos y antónimos, mejorar el análisis de sentimientos, la corrección de las faltas de ortografía y cualquier acople que pueda hacer la experiencia de desarrollar una obra toda una experiencia sin tener que moverse de la plataforma para consultar o recibir retroalimentación de las obras escritas.

9

Conclusión

Para la realización de este proyecto los dos integrantes del equipo tenían como objetivo principal aplicar y llevar a la práctica todos los conceptos aprendidos a lo largo del grado de ingeniería de software junto con aprender nuevas tecnologías y lenguajes para obtener un conocimiento más amplio acerca del ámbito del desarrollo de aplicaciones web.

Como se ha ido explicando a lo largo de todos los capítulos, se decidió desarrollar una aplicación full-stack donde las tecnologías candidatas a usarse en cada ámbito fueron varias. En el ámbito del front-end sonaron varias como React, Vue.js, Angular, siendo esta última como la elegida. En el ámbito del back-end no hubo mucho debate ya que el reciente trabajo con las tecnologías .Net han sido de gran agrado y se quiso ampliar los conocimientos en este nicho, por lo tanto, .Net Core fue la tecnología elegida para la construcción del back-end. Para el ámbito de almacenar la información que mostraría la aplicación hubo debate sobre si usar una base de datos relacional, no relacional, orientada a documentos, etc, en un principio se pensó en usar MongoDB directamente, pero salió la opción de poder usar una base de datos en nube de Azure con el esquema de MongoDB, por lo tanto, la elegida fue CosmosDB.

Todas estas elecciones fueron tomadas pensando en los beneficios que podían aportar tanto personal como profesionalmente, ya que son tecnologías que actualmente tienen una gran demanda profesional en el sector de las tecnologías.

La etapa más problemática del proyecto se encontró al principio cuando se intentaba montar una solución escalable y con unas capas bien definidas que facilitase la vida a desarrollar día a día y en el momento de probar los desarrollos no provocase confusión o algún problema. Crear el proyecto con el SDK de Microsoft en la versión .Net Core 2.0 dio ciertos problemas con Angular 5 y Bootstrap 4 pero se consiguió resolver actualizando el SDK de .Net Core a la versión 2.1. Los problemas que se encontraron más adelante se resolvieron gracias a que hay una gran comunidad detrás de las tecnologías seleccionadas y fueron de gran ayuda para poder hacer frente a problemas de comunicación entre las tecnologías entre otros.

El desarrollo del proyecto con dichas tecnologías ha sido gratificante y ha mantenido un ritmo de aprendizaje alto al no conocer las tecnologías con las que se decidió desarrollar el proyecto.

Gracias a la parte de pruebas realizadas en la parte de back-end se detectó algunos errores y permitió eliminar parte de código que estaba repetido y se podía generalizar para tener una solución más limpia y entendible.

Un punto muy positivo ha sido que el proyecto sea un trabajo en grupo ya que ha dado pie a muchos debates y enseñanzas por todas las partes en la toma de decisiones en la forma de construir y diseñar la solución, planificación de tareas, detección y reportes de errores en el desarrollo y como hacer frente a los problemas que surgían a medida que iba avanzando el proyecto.

Gracias al desarrollo de este proyecto al final se han obtenido unas habilidades en el desarrollo de la parte front-end de aplicaciones que al principio no se poseía, llegando a entender un poco más el paradigma actual. En cuanto a la tecnología del back-end ha permitido ampliar y mejorar el diseño de una Api Rest y en su caso las pruebas de integración para comprobar que funciona correctamente.

También se han adquirido conocimientos en el aspecto de bases de datos no relacionales ya que al principio no se conocía ni se había trabajado con ninguna base de datos NoSQL. Esto ha hecho que el desarrollo de la aplicación haya sido de forma paulatina a la par que iba desarrollándose el modelo de datos.

Finalmente, la experiencia aprendiendo y aplicando todos los conceptos aquí vistos ha sido fructífera y muy provechosa aportando una experiencia valiosa ampliando la visión del panorama actual del desarrollo de aplicaciones y más concretamente el desarrollo web.

Referencias

¿Qué es NuGet y qué hace? (08 de enero de 2019). Obtenido de Microsoft

Docs: <https://docs.microsoft.com/es-es/nuget/what-is-nuget>

Angular - Introduction to the Angular Docs. (10 de enero de 2019). Obtenido de

Angular Docs: <https://angular.io/docs>

Angular | Font Awesome. (16 de enero de 2019). Obtenido de Font Awesome:

<https://fontawesome.com/how-to-use/on-the-web/using-with/angular>

Angular powered Bootstrap. (04 de marzo de 2019). Obtenido de ng-bootstrap:

<https://ng-bootstrap.github.io/#/home>

API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos. (18 de diciembre de 2018). Obtenido de BBVA API Market:

<https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>

Azure Cosmos DB: servicio de base de datos multimodelo. (03 de diciembre de 2018). Obtenido de Microsoft Azure: <https://azure.microsoft.com/es-es/services/cosmos-db/>

Blanco, N. (21 de marzo de 2019). *¿Qué patron usa Angular? MVC o MVVM.*

Obtenido de OpenWebinars: <https://openwebinars.net/blog/que-patron-usa-angular-mvc-o-mvvm/>

Chacon, S. (12 de enero de 2019). *Git.* Obtenido de Git: <https://git-scm.com/>

Chart.js | Open source HTML5 for your website . (01 de abril de 2019).

Obtenido de Chart.js: <https://www.chartjs.org/>

Documentación de ASP.NET. (15 de noviembre de 2018). Obtenido de

Microsoft: <https://docs.microsoft.com/es-es/aspnet/?view=aspnetcore-2.2#pivot=core>

Documentation for Visual Studio Code. (10 de enero de 2019). Obtenido de

Visual studio code: <https://code.visualstudio.com/docs>

Explicación de DevOps: Información general sobre las tecnologías de DevOps.

(s.f.). Obtenido de Microsoft Azure: <https://azure.microsoft.com/es-es/overview/devops/>

Guía de C#. (02 de octubre de 2018). Obtenido de Microsoft Docs:

<https://docs.microsoft.com/es-es/dotnet/csharp/>

HMACSHA512 Class. (25 de Enero de 2019). Obtenido de Microsoft Docs:

<https://docs.microsoft.com/es-es/dotnet/api/system.security.cryptography.hmacsha512?view=netcore-2.1>

Introducción a las consultas LINQ (C#). (18 de diciembre de 2018). Obtenido de Microsoft Docs: <https://docs.microsoft.com/es-es/dotnet/csharp/programming-guide/concepts/linq/introduction-to-linq-queries>

Introduction Bootstrap. (11 de enero de 2019). Obtenido de Bootstrap:

<https://getbootstrap.com/docs/4.3/getting-started/introduction/>

JSON. (05 de enero de 2019). Obtenido de Wikipedia:

<https://es.wikipedia.org/wiki/JSON>

JSON Web Tokens. (05 de enero de 2019). Obtenido de JWT: <https://jwt.io/>

Json.NET. (19 de enero de 2019). Obtenido de Newtonsoft:

<https://www.newtonsoft.com/json>

MongoDB: The most popular database for modern apps. (04 de febrero de 2019). Obtenido de MongoDB: <https://www.mongodb.com/>

npm Documentation. (25 de enero de 2019). Obtenido de npm Documentation:

<https://docs.npmjs.com/>

Quickstart - Quill Rich Text Editor. (02 de febrero de 2019). Obtenido de Quilljs:

<https://quilljs.com/docs/quickstart/>

Scrum (desarrollo de software). (10 de marzo de 2019). Obtenido de Wikipedia:

[https://es.wikipedia.org/wiki/Scrum_\(desarrollo_de_software\)](https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software))

Spec India. (14 de enero de 2019). *React vs Angular vs Vue.js: A Complete Comparison Guide*. Obtenido de Medium: <https://medium.com/front-end-weekly/react-vs-angular-vs-vue-js-a-complete-comparison-guide-d16faa185d61>

SWAGGER Y SWAGGER UI: ¿QUÉ ES Y POR QUÉ ES IMPRESCINDIBLE PARA TUS APIS? (s.f.). Obtenido de Chakray:

<https://www.chakray.com/es/swagger-y-swagger-ui-por-que-es-imprescindible-para-tus-apis/>

Transferencia de Estado Representacional. (18 de noviembre de 2018).

Obtenido de Wikipedia:

https://es.wikipedia.org/wiki/Transferencia_de_Estado_Representacional

TypeScript. (10 de enero de 2019). Obtenido de Wikipedia:

<https://es.wikipedia.org/wiki/TypeScript>

Valor Software. (05 de marzo de 2019). *NGX Bootstrap*. Obtenido de NGX

Bootstrap: <https://valor-software.com/ngx-bootstrap/#/documentation>

Visual Studio Documentación. (01 de octubre de 2018). Obtenido de Microsoft

Docs: <https://docs.microsoft.com/es-es/visualstudio/?view=vs-2017>

Anexo: Guía de uso para usuario

En este apartado se explicarán los pasos que los usuarios han de seguir para usar la aplicación web, adjuntando para ello imágenes que ilustren y clarifiquen el recorrido por la misma.

Acceso a la aplicación

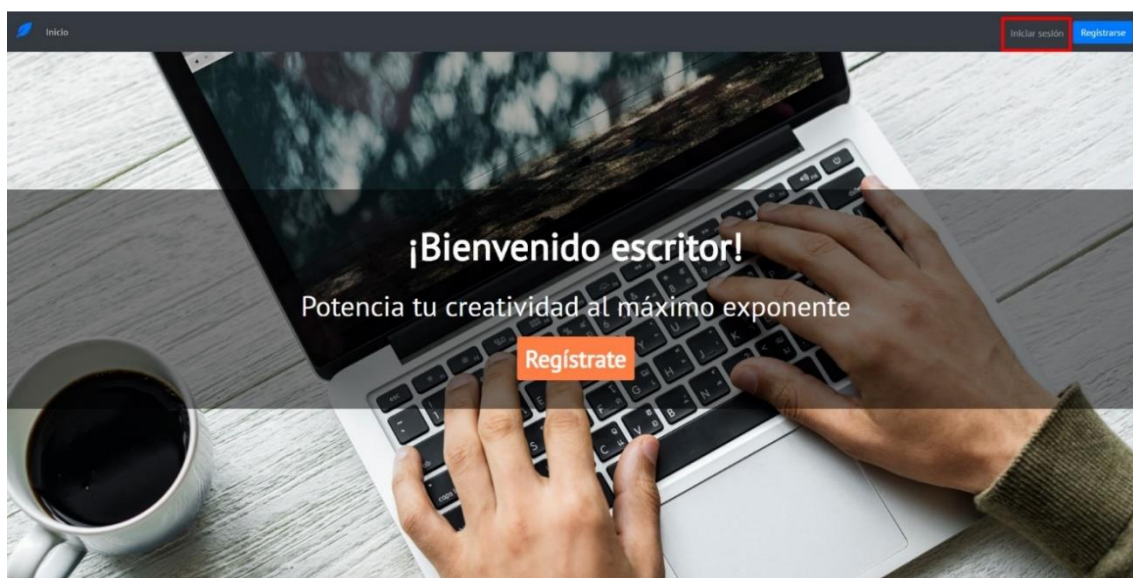


Figura 63 – Página de bienvenida a la aplicación

Para usuarios registrados: en la pestaña de inicio, pulsar el botón “Iniciar sesión”.

Entonces se abre una ventana donde introducir los datos de acceso: correo electrónico y contraseña, ambos obligatorios. Tras introducirlos, pulsar el botón “Iniciar sesión”.

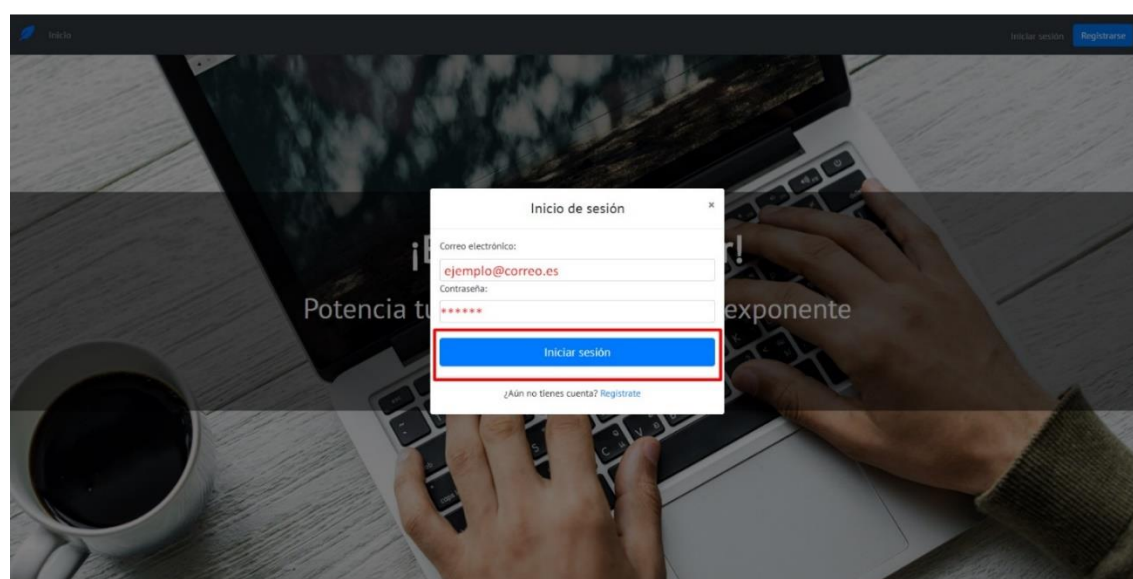


Figura 64 – Página de inicio de sesión

Para usuarios no registrados: en la pestaña de inicio, pulsar alguno de los dos botones de registro.

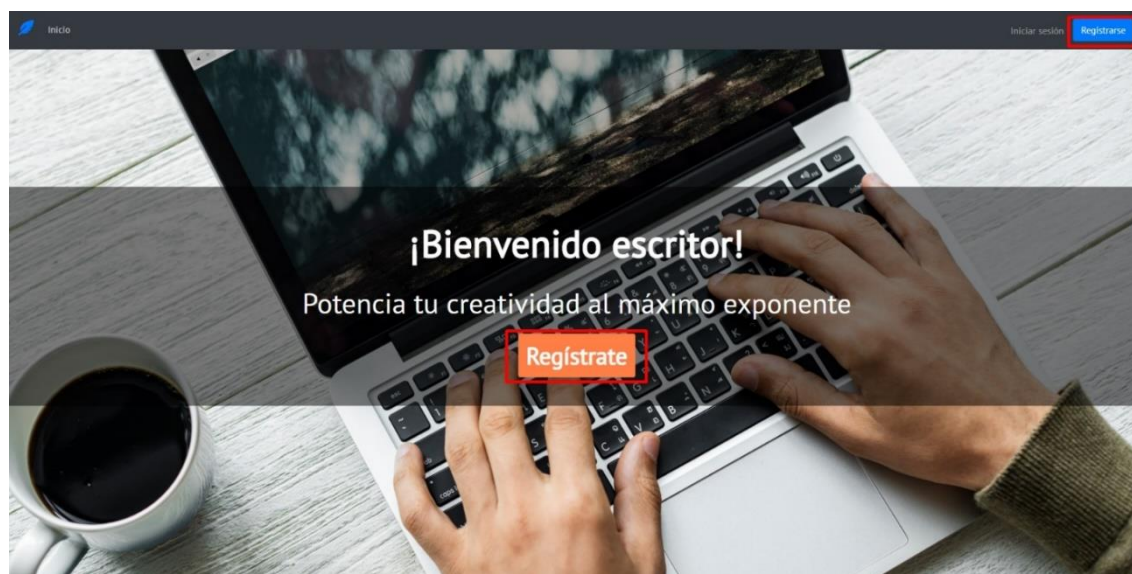


Figura 65 – Página de bienvenida, botones para registrarse

Se abre ventana de registro, todos los campos son obligatorios, y el botón “registrarme” aparece desactivado hasta que se rellenen los campos.

Una vez completados, pulsar el botón. Tras ello, deberá iniciar sesión introduciendo los datos de accesos “correo electrónico” y “contraseña”.

Figura 66 – Página de registro de usuarios

Cerrar sesión

Una vez se accede a la aplicación, arriba a la derecha en la barra de navegación, aparece un desplegable, al pulsar, se visualiza la opción de “cerrar sesión”. Pulsar sobre ella.

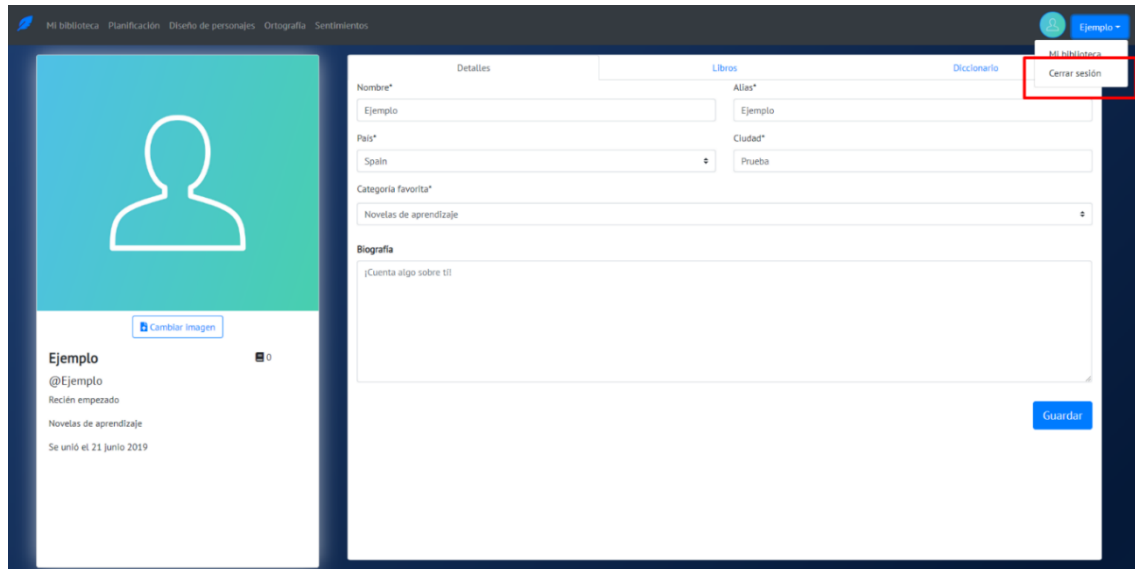


Figura 67 – Acción de cerrar sesión de la aplicación

Editar datos del perfil

En la pestaña de mi biblioteca, se visualiza los datos del perfil del escritor junto a sus libros. Pulsar sobre detalles y rellenar los datos que se desean modificar. No se puede dejar en blanco ninguno que esté marcado con asterisco.

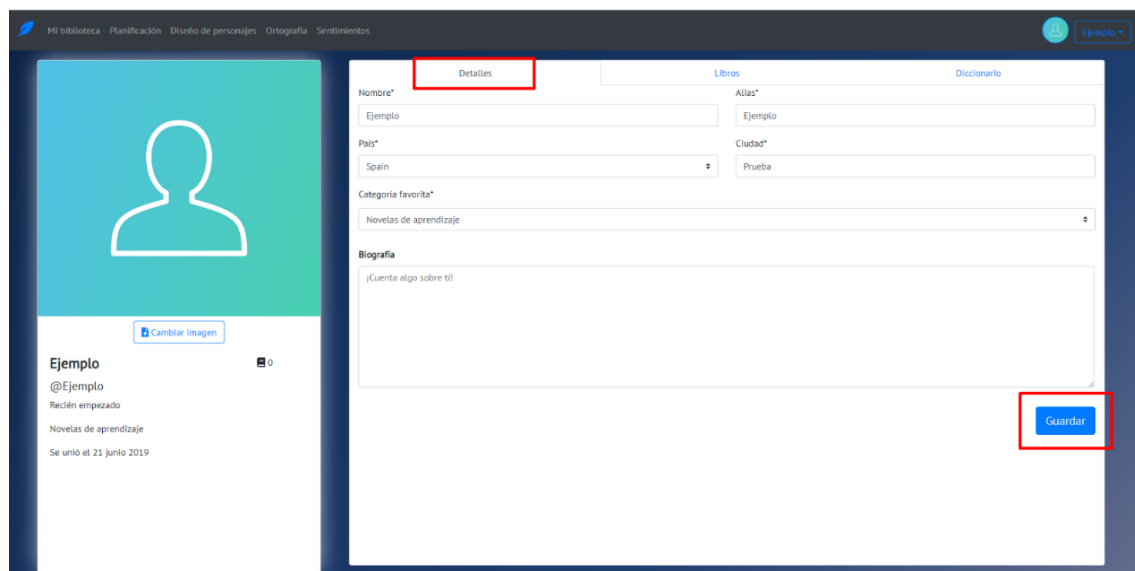


Figura 68 – Perfil del escritor, detalles del escritor

Si se desea cambiar la imagen del perfil, pulsar sobre “Cambiar imagen”.

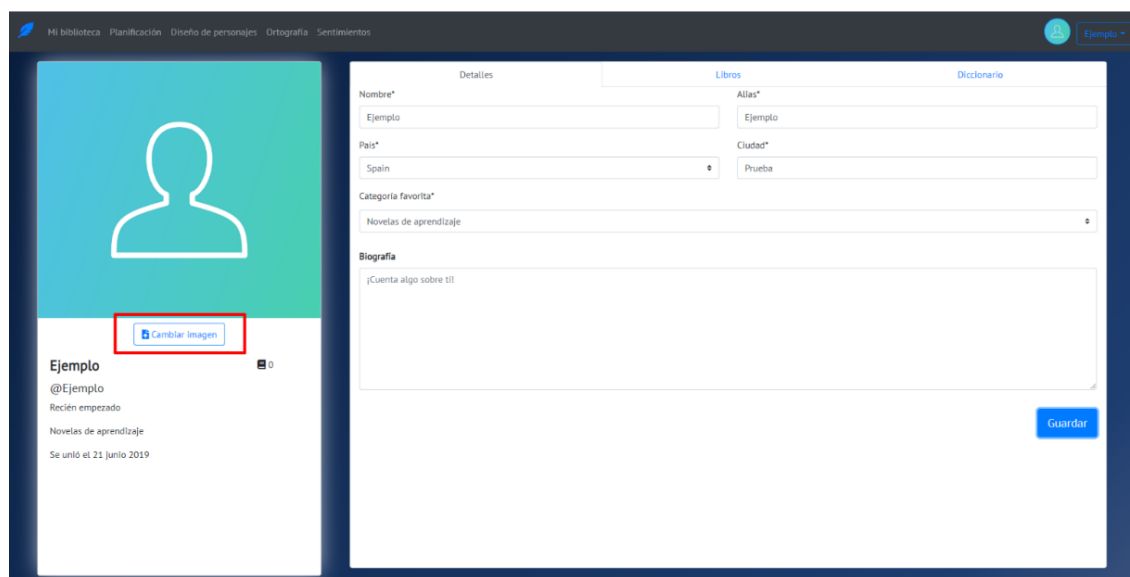


Figura 69 – Subida de imagen en el perfil del escritor

Al hacerlo se abrirá el explorador de archivos del sistema operativo. Seleccionar una imagen.

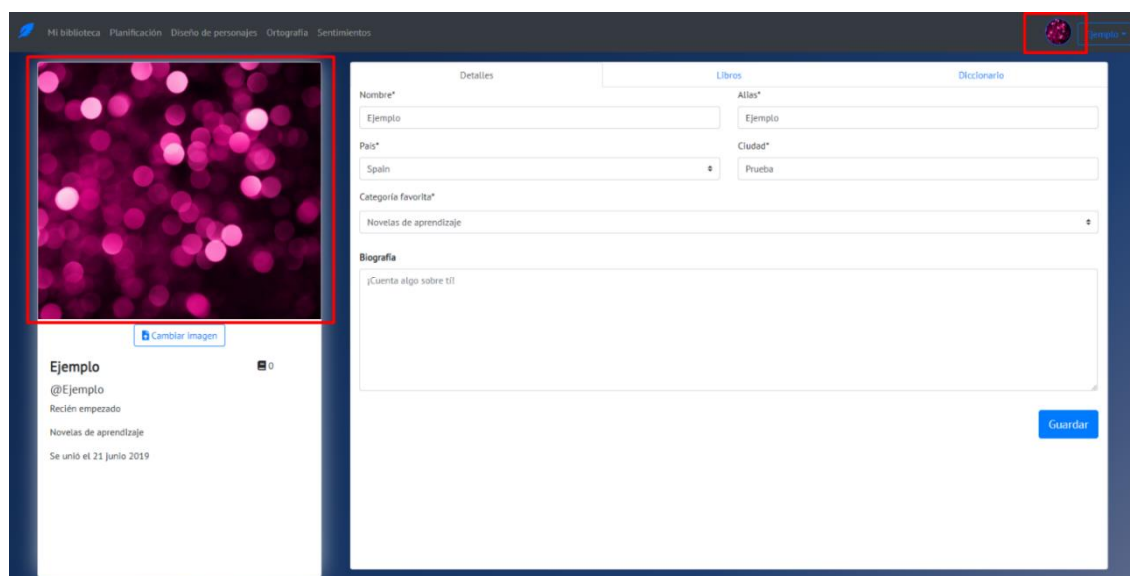


Figura 70 – Perfil del usuario con imagen cambiada

Planificación de tareas

Seleccionar “Planificación” en la barra de navegación.

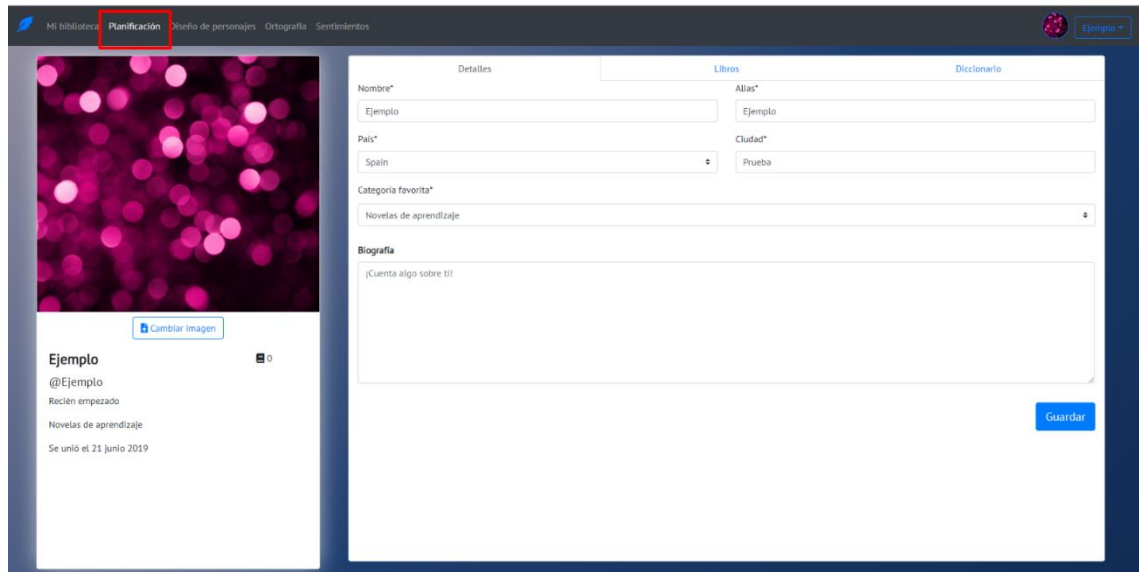


Figura 71 – Navegar desde perfil del escritor al planificador de código

Una vez dentro, crear una tarea pulsando el botón +

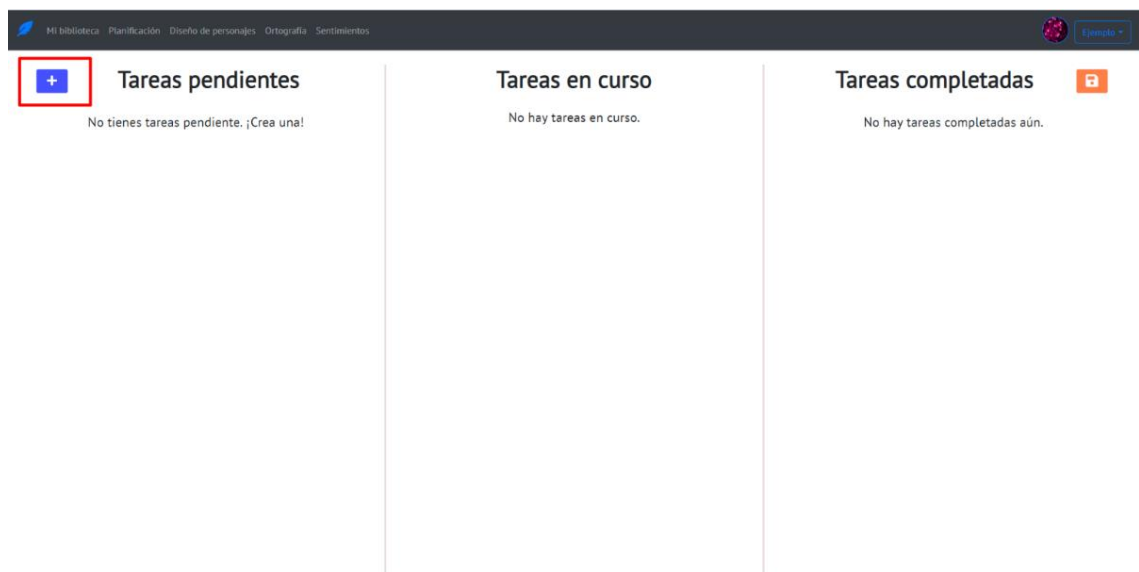


Figura 72 – Página de planificador de tareas

Se abrirá una ventana para rellenar el título de la tarea, la descripción y la fecha esperada de cumplimiento. El título y la fecha son obligatorios.

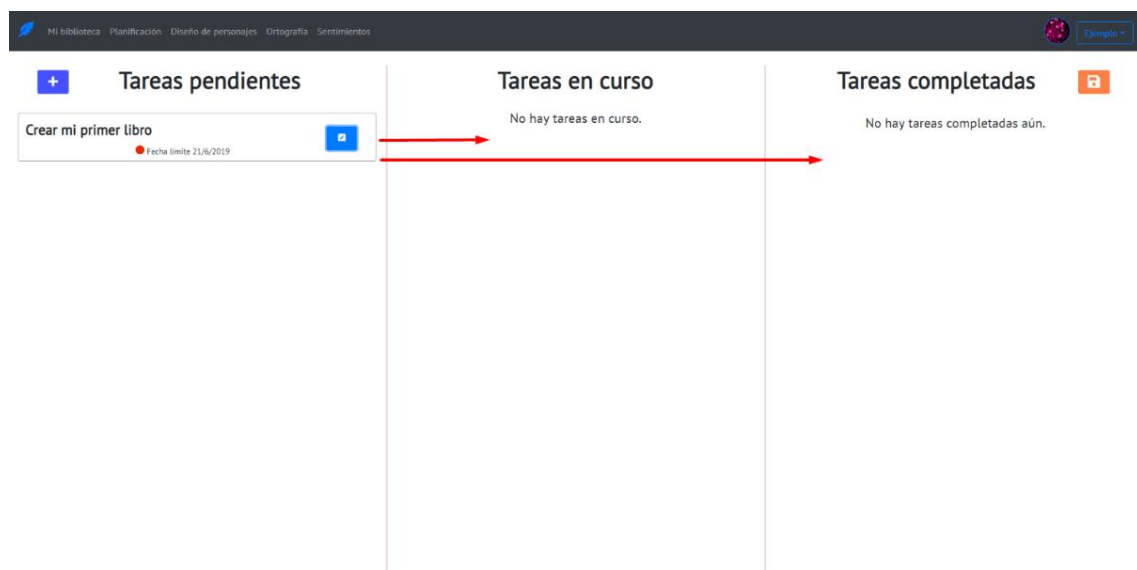


Figura 73 – Planificador de tareas, acción de mover tarea pendiente
Se creará en tareas pendientes, y podrá ser arrastrada a las demás columnas.

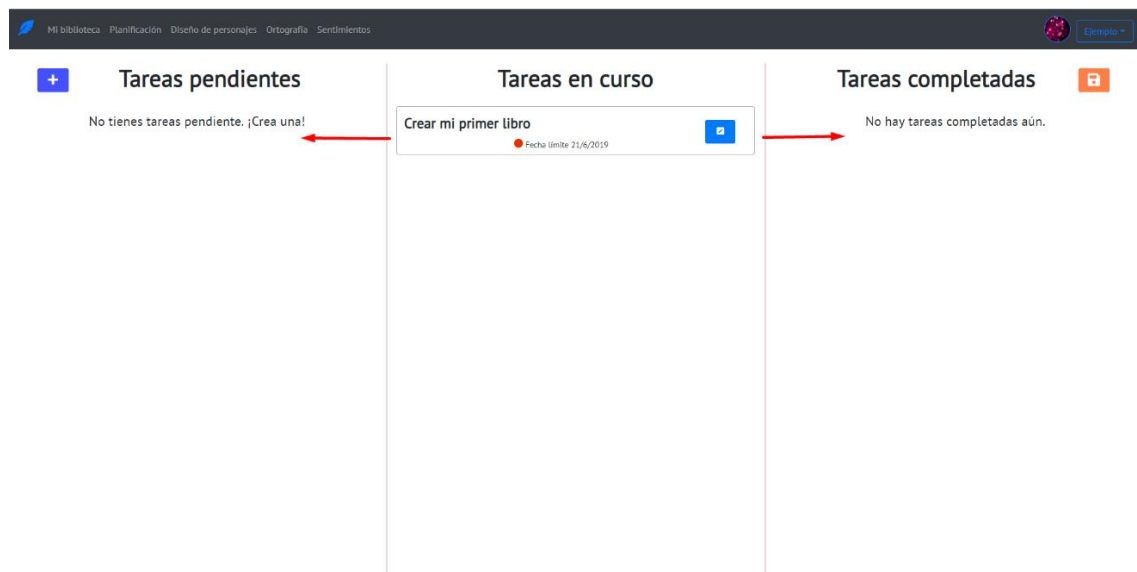


Figura 74 – Planificador de tareas, acción de mover tarea en curso



Figura 75 – Planificador de tareas, acción de mover tarea completada

Para editar el estado de la tarea pulsar el botón de edición.

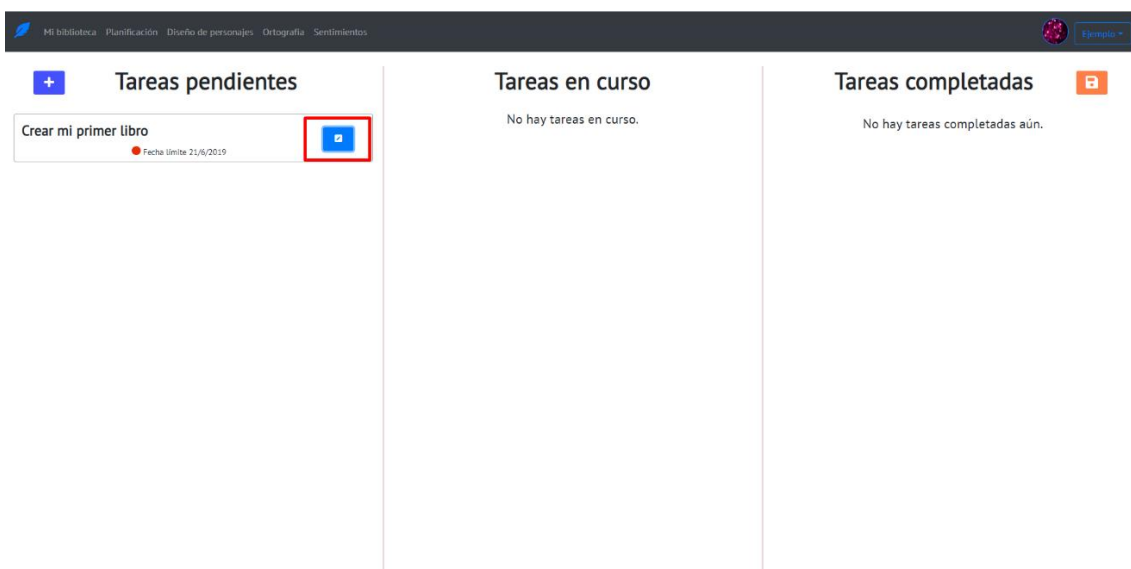


Figura 76 – Planificador de tareas, acción de mover tarea en curso

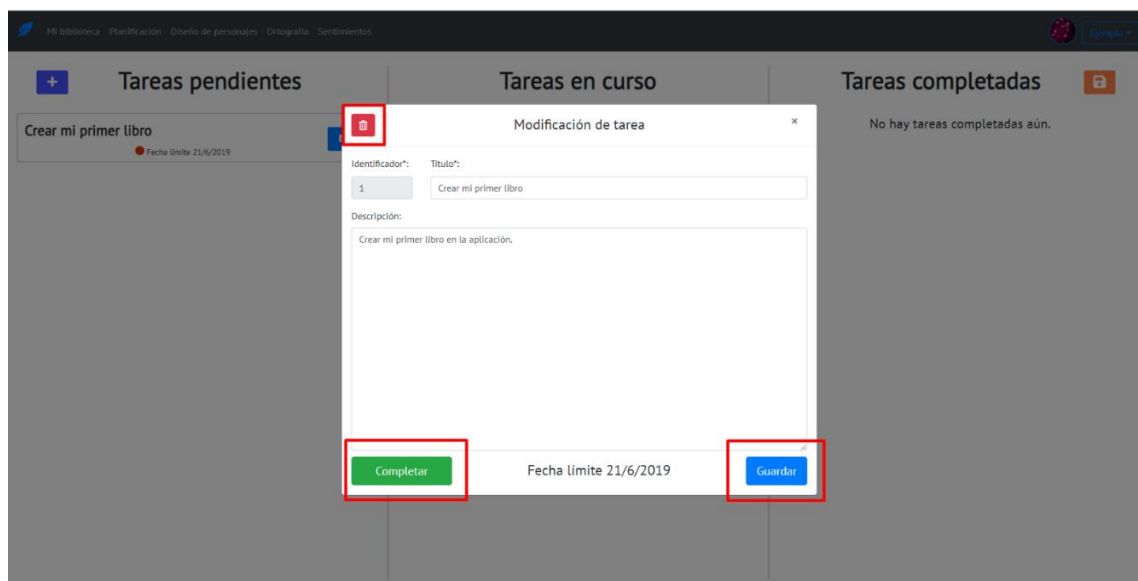


Figura 77 – Planificador de tareas, modificar tarea

Tras pulsar, se visualiza una ventana de edición. Se podrá borrar, editar, o completar la tarea.

Al pulsar en el botón de la papelera se abre una ventana de confirmación para borrar, seleccionar “Eliminar” si realmente se desea borrar la tarea.

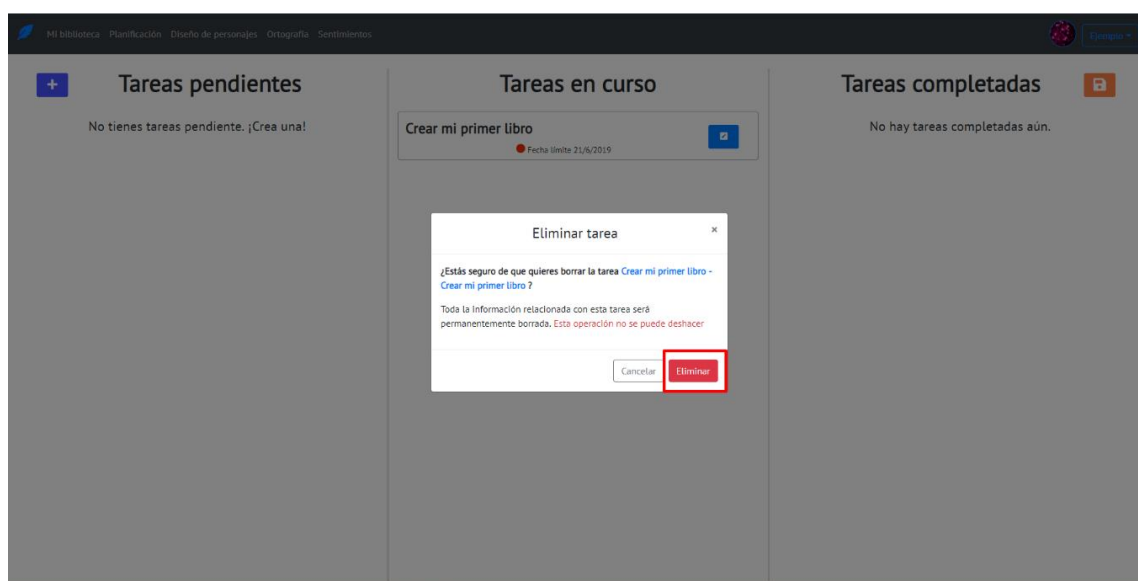


Figura 78 – Planificador de tareas, acción de eliminar tarea

Pulsar el botón “Guardar”, para confirmar los nuevos cambios sobre la tarea.

Pulsar el botón “Completar” si lo que desea es poner la tarea en la columna “Tareas completadas”.

Para que no se pierdan los cambios y el estado final del panel de tareas, pulsar sobre el botón de guardado de arriba a la derecha.

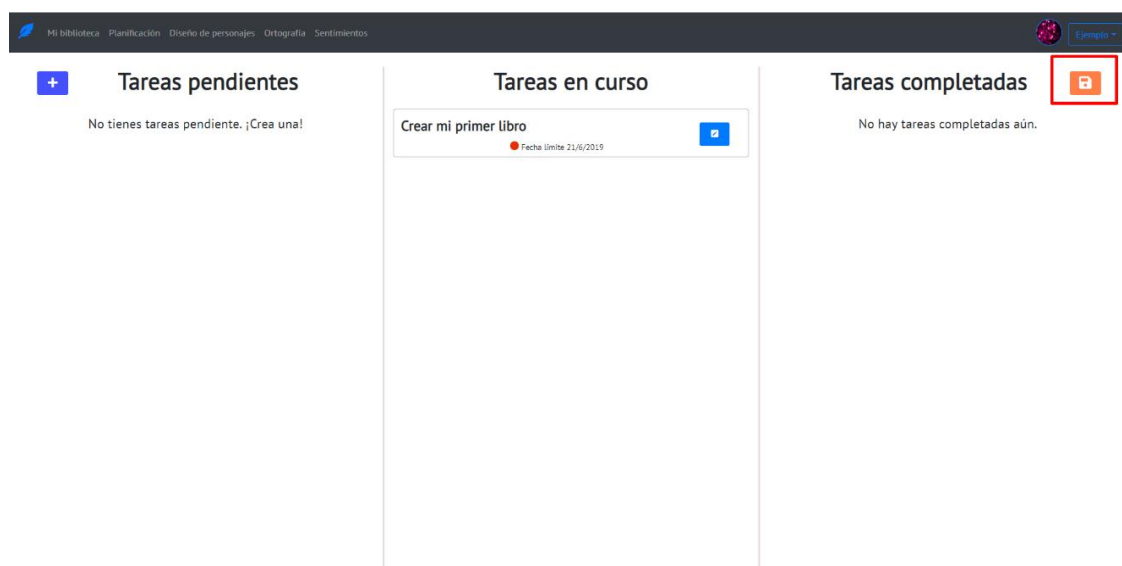


Figura 79 – Planificador de tareas, acción de guardar cambios

Crear, borrar editar palabra en el diccionario

Desde la pantalla Mi biblioteca, seleccionar la pestaña diccionario, desde ahí, pulsar “Nueva palabra”.

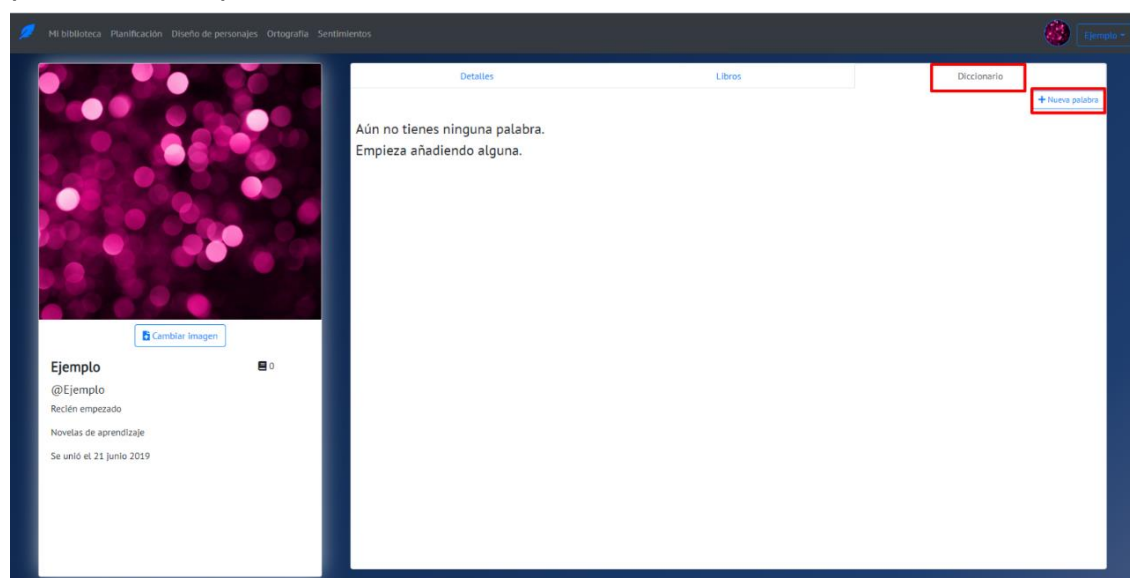


Figura 80 – Página del diccionario del escritor

Al hacerlo aparece una ventana de creación donde rellenar la palabra y su descripción. El campo “palabra” es obligatorio y hasta que no se rellene no se podrá crear la palabra.

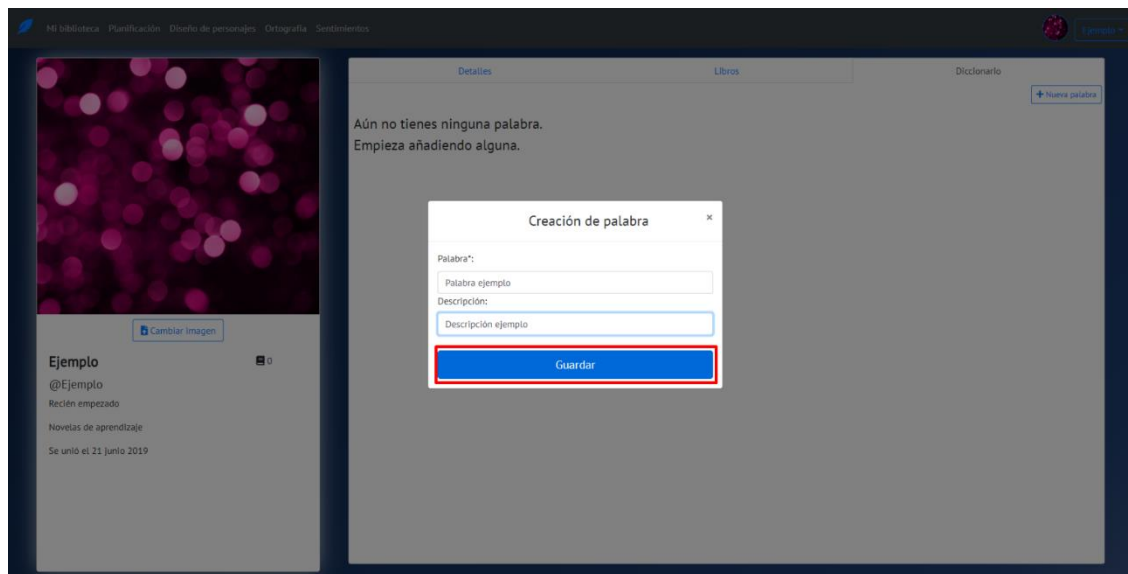


Figura 81 – Formulario de creación de palabra del diccionario

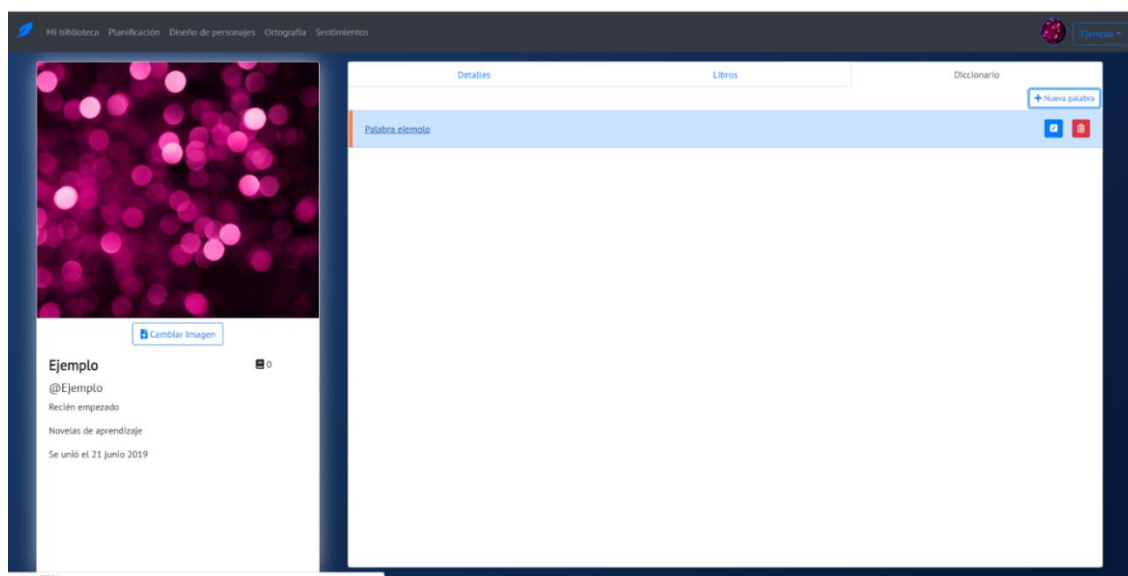


Figura 82 – Página del diccionario del escritor con palabra añadida

Para visualizar su descripción, pulsar sobre la palabra en la lista.

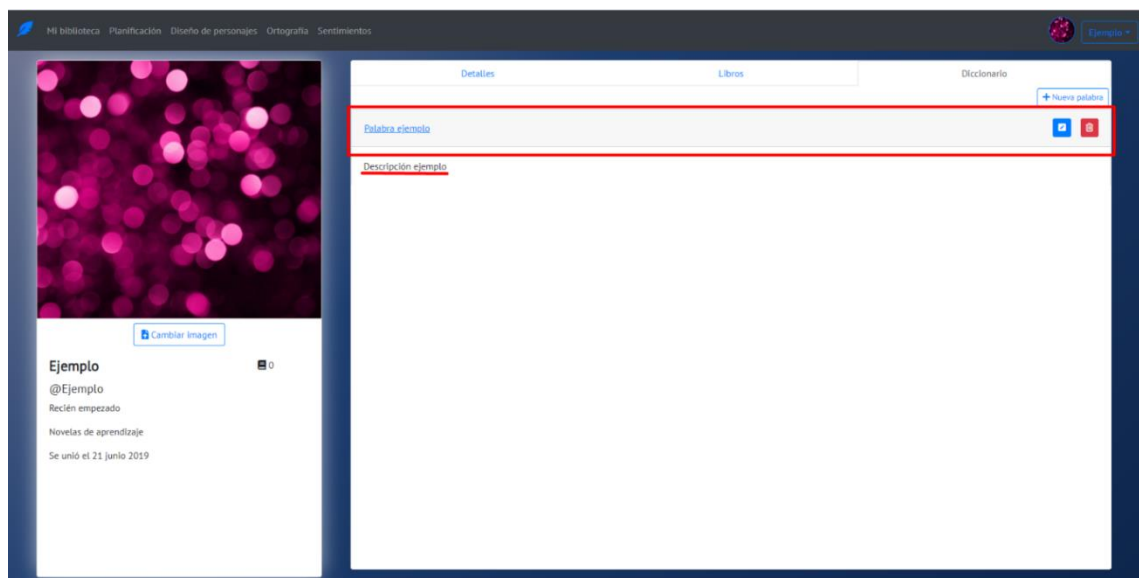


Figura 83 – Diccionario del escritor acción pinchar sobre palabra

Para editar, o borrar, pulsar los botones situados a la derecha respectivamente.

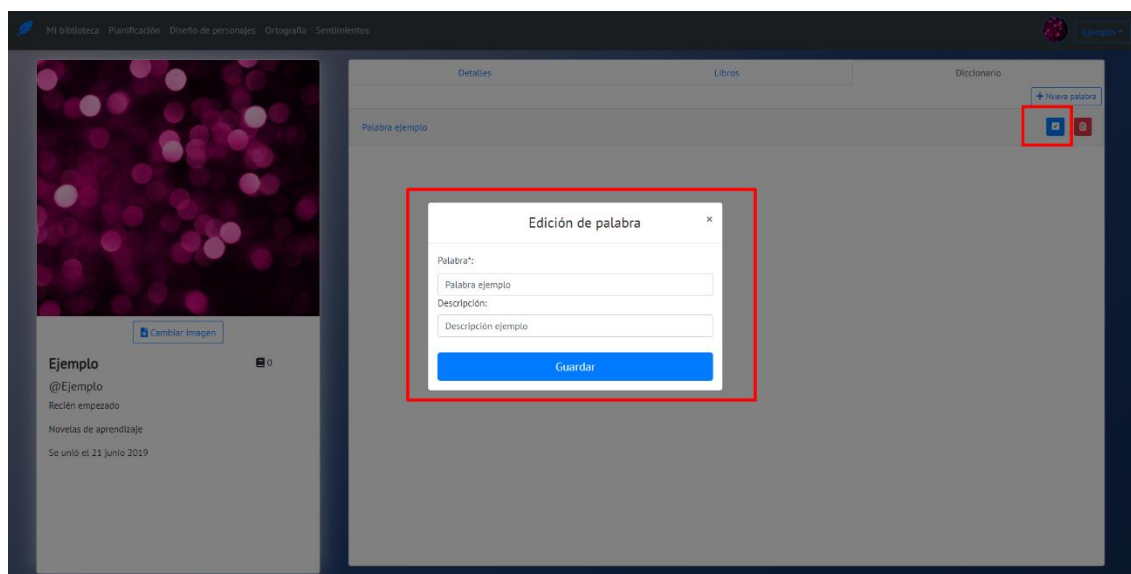


Figura 84 – Diccionario del escritor, formulario de modificación de palabra

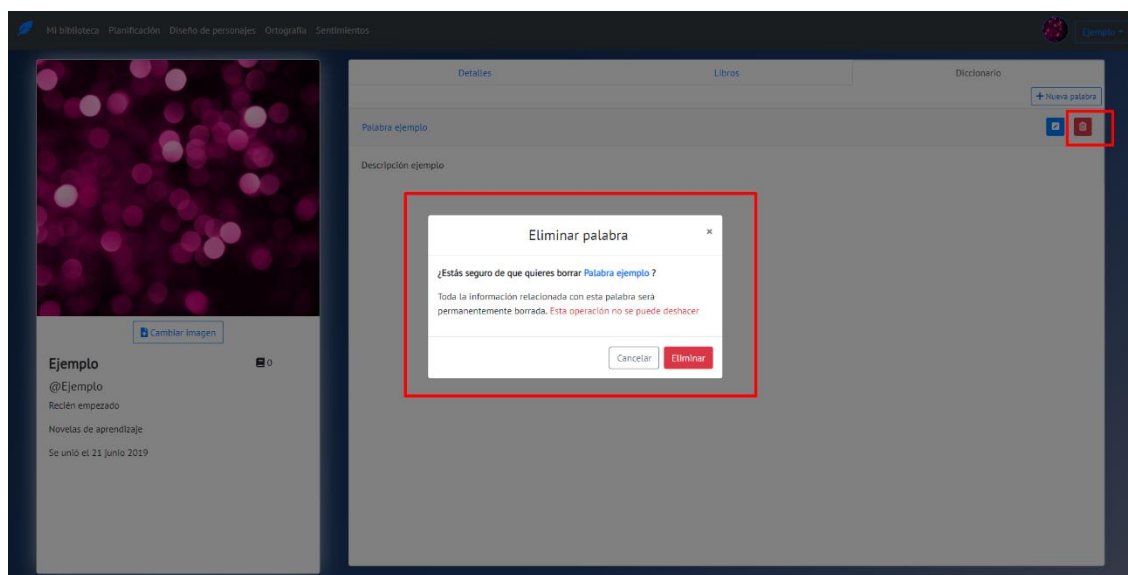


Figura 85 – Acción de eliminar una palabra del diccionario

Crear y borrar libro

Desde la pantalla de mi biblioteca, seleccionar la pestaña libros y pulsar “Nuevo libro”.

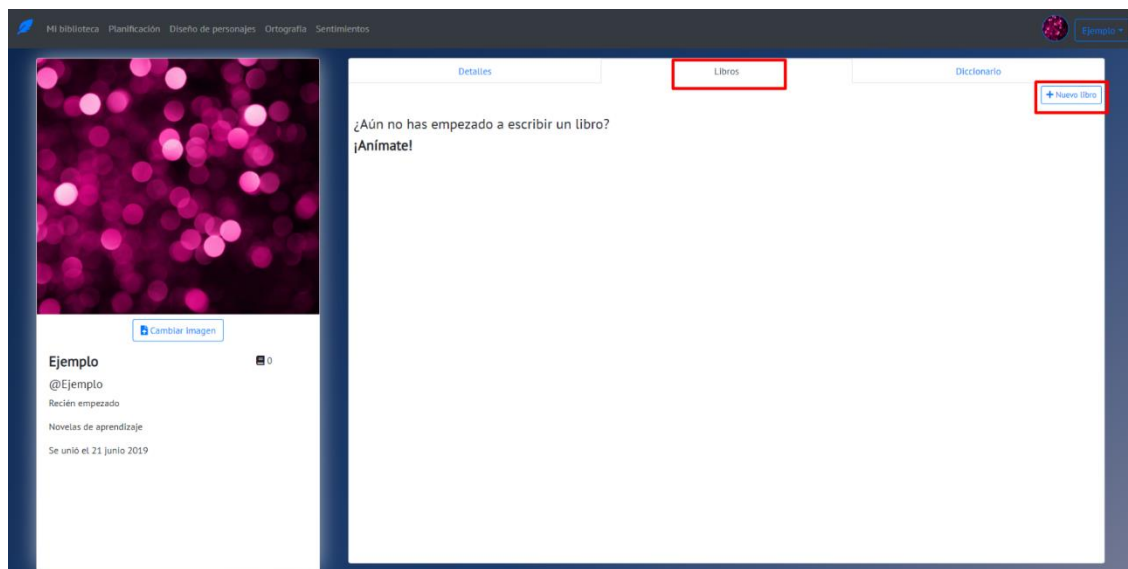


Figura 86 – Página del listado de libros del escritor

Al hacerlo aparece la ventana de creación. Ningún campo es obligatorio, se puede crear el libro al pulsar el botón crear libro.

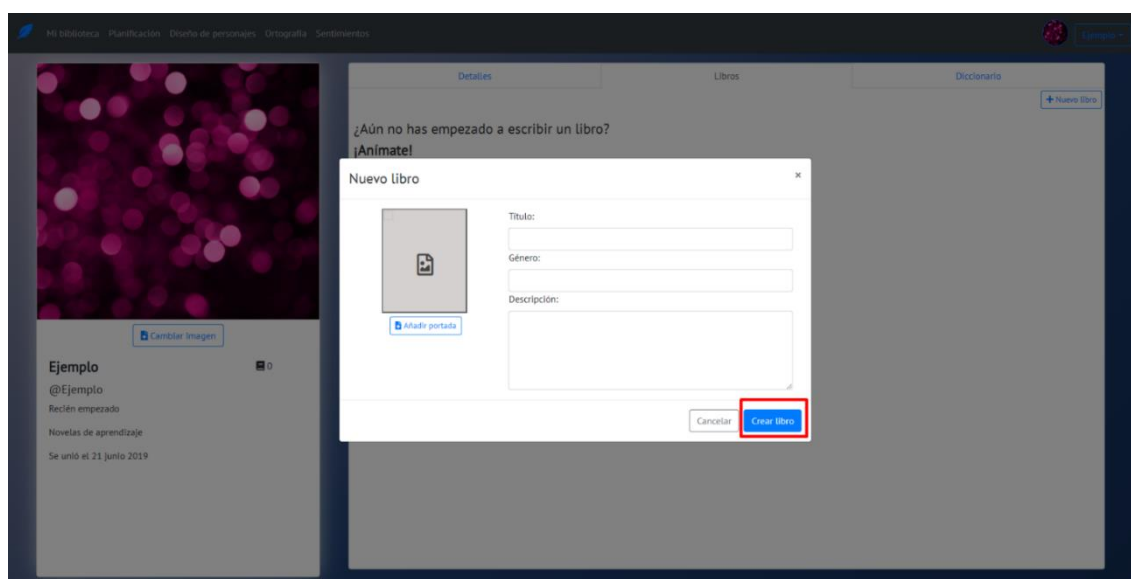


Figura 87 – Página creación de libro nuevo

Si se desea subir la portada al momento de creación, pulsar “Añadir portada”. Se abre explorador de archivos y se podrá elegir una imagen. Al seleccionarla se visualiza.

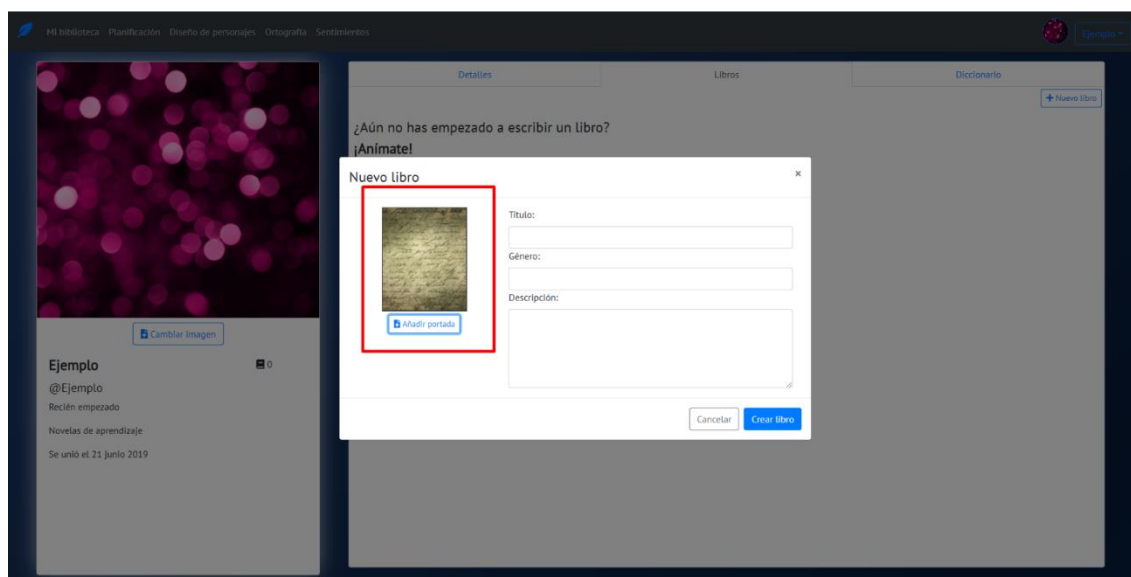


Figura 88 – Subida de imagen de la portada del libro

Al pulsar botón “Crear libro”, se accede directamente a su pantalla de detalles. Otra forma de acceder es pulsando sobre el libro creado en la lista.

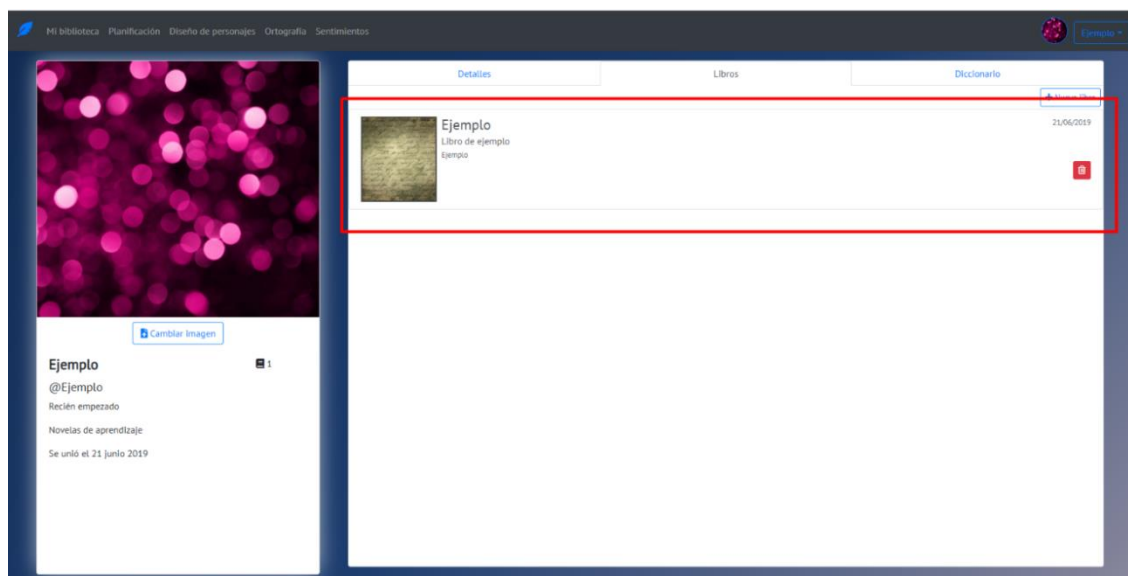


Figura 89 – Página del listado de libros

Si se desea borrar el libro, pulsar el botón rojo de la papelera y dar al botón de confirmar una vez aparezca la pestaña de confirmación.

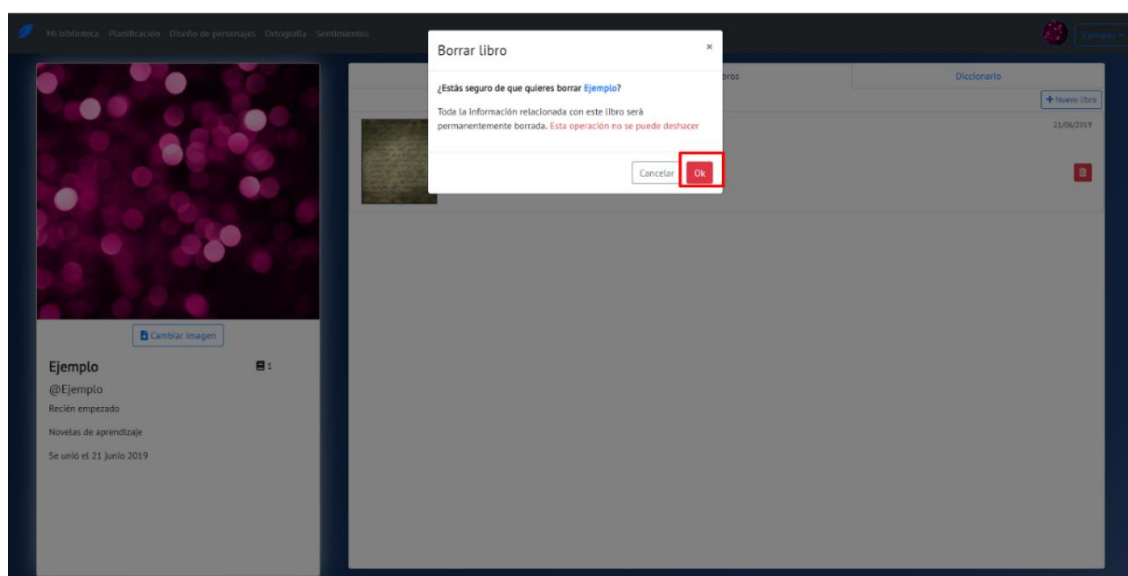


Figura 90 – Página de la eliminación de un libro

Editar detalles de libro

Desde la pantalla de detalles del libro, pulsar la pestaña “Detalles”. Ningún campo es obligatorio, rellenarlos y pulsar el botón “Guardar”. Si se desea cambiar la imagen de la portada, pulsar sobre el botón “Cambiar portada”, y seleccionar imagen.

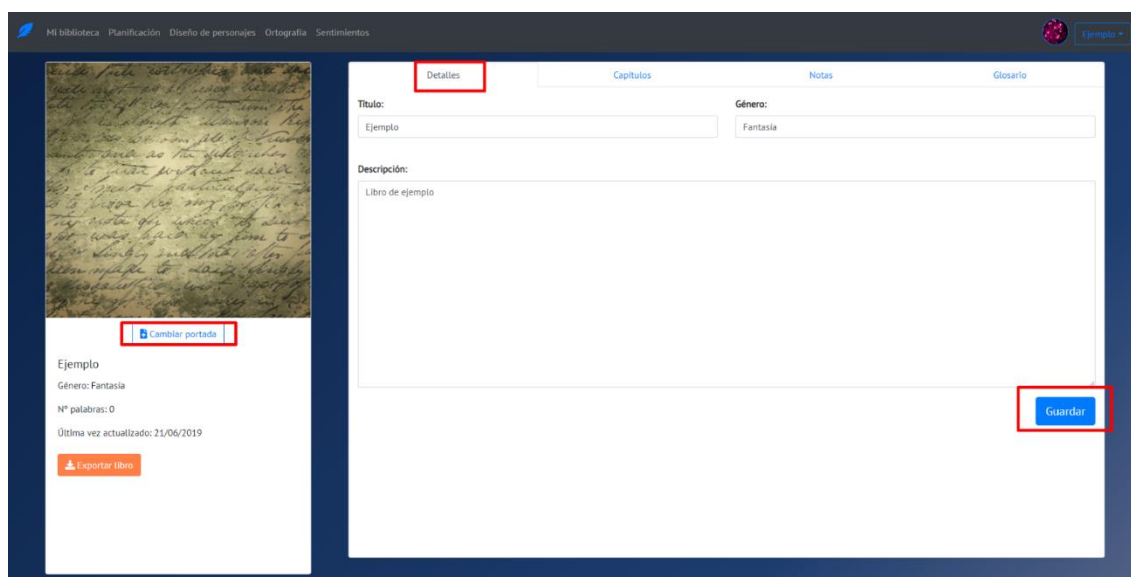


Figura 91 – Página de detalles del escritor

Crear, editar y borrar palabra glosario

Este apartado tiene el mismo funcionamiento que la creación de palabras del diccionario personal del escritor. Acceder a la pestaña correspondiente, “Glosario”, seleccionar “Nueva palabra”, pulsar sobre ella una vez creada para ver el significado, y pulsar sobre los botones de edición o borrado en caso de necesitarlo.

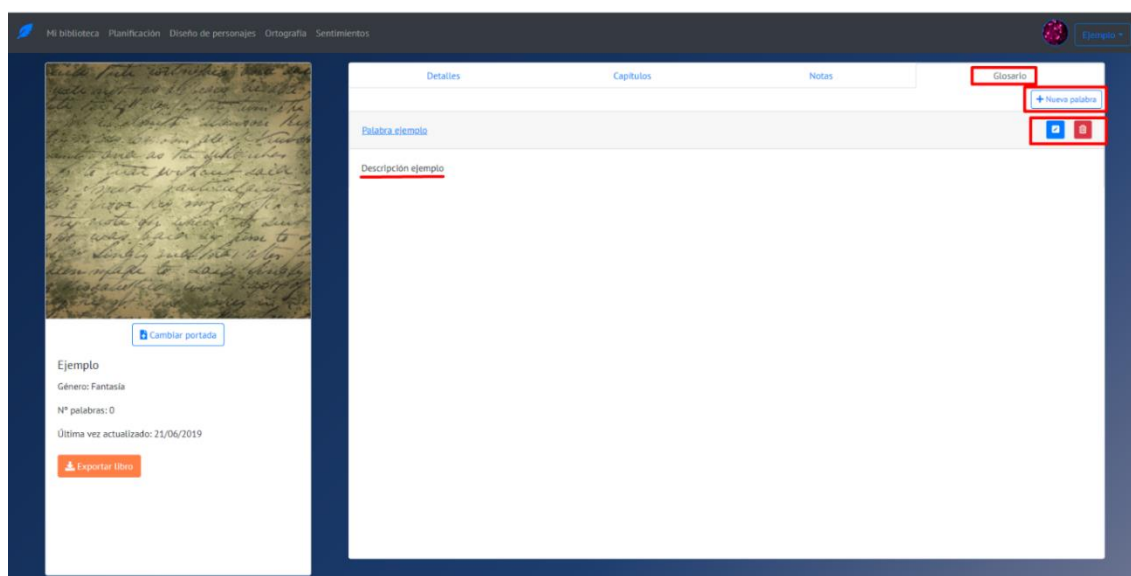


Figura 90 – Página del glosario de un libro

Crear editar y borrar nota

De nuevo, el conjunto de pasos es el mismo que los vistos en diccionario y glosario, la única diferencia es que no hay botón de editado, basta con pulsar sobre el elemento una vez creado, ya que la información se visualiza desde fuera.

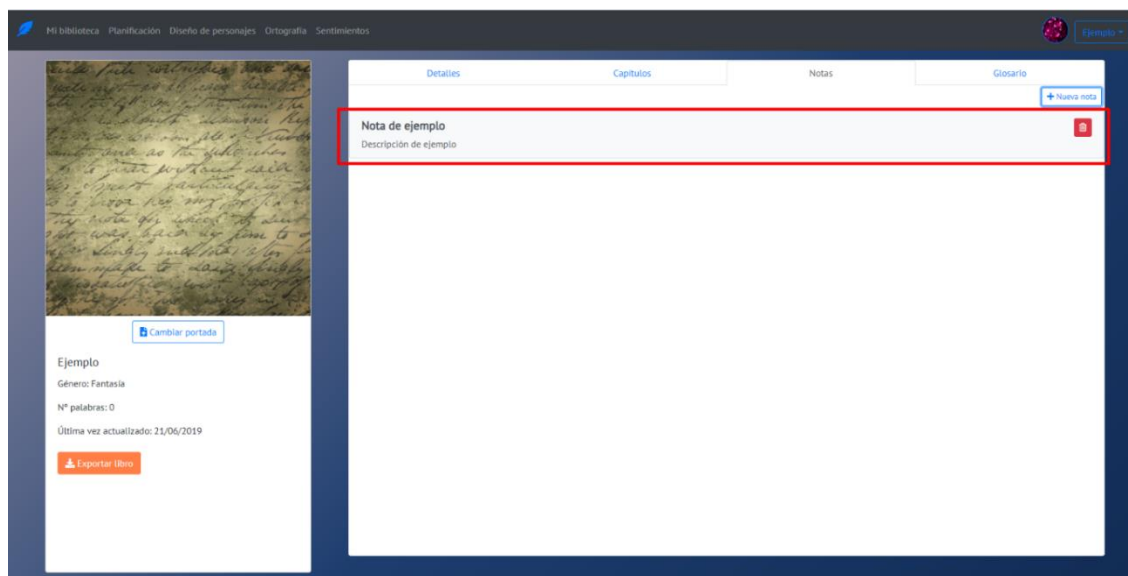


Figura 91 – Página del listado de notas de un libro

Si la descripción es demasiado larga, se puede ajustar el tamaño de entrada, arrastrando desde la esquina inferior derecha el recuadro de texto.

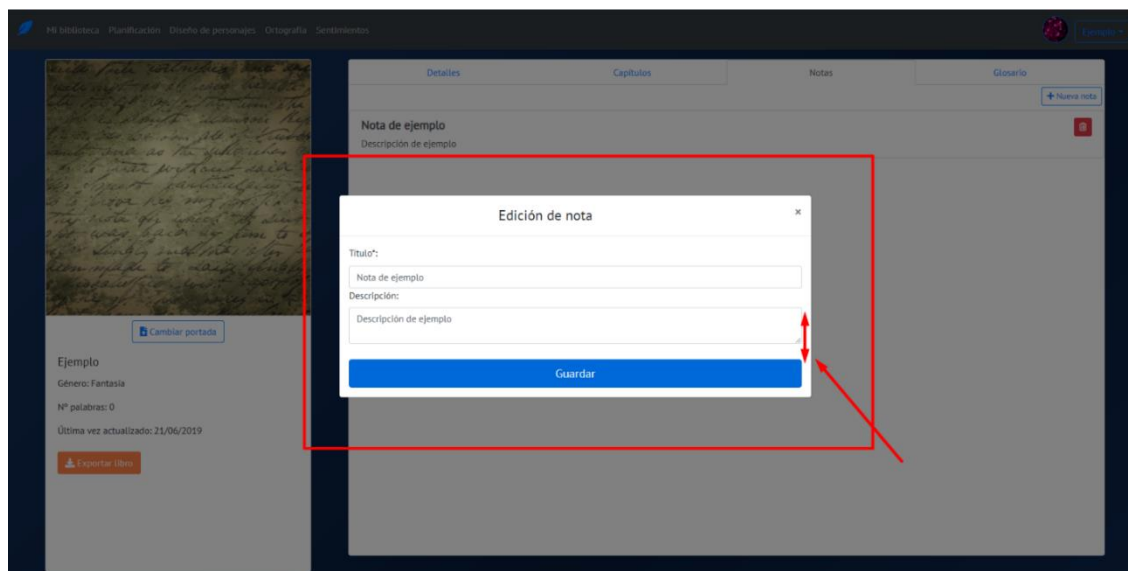


Figura 92 – Página de modificación de una nota

Crear y borrar capítulos

Desde la pantalla de detalles del libro, pulsar la pestaña “Capítulos”, y pulsar el botón “Nuevo Capítulo”.

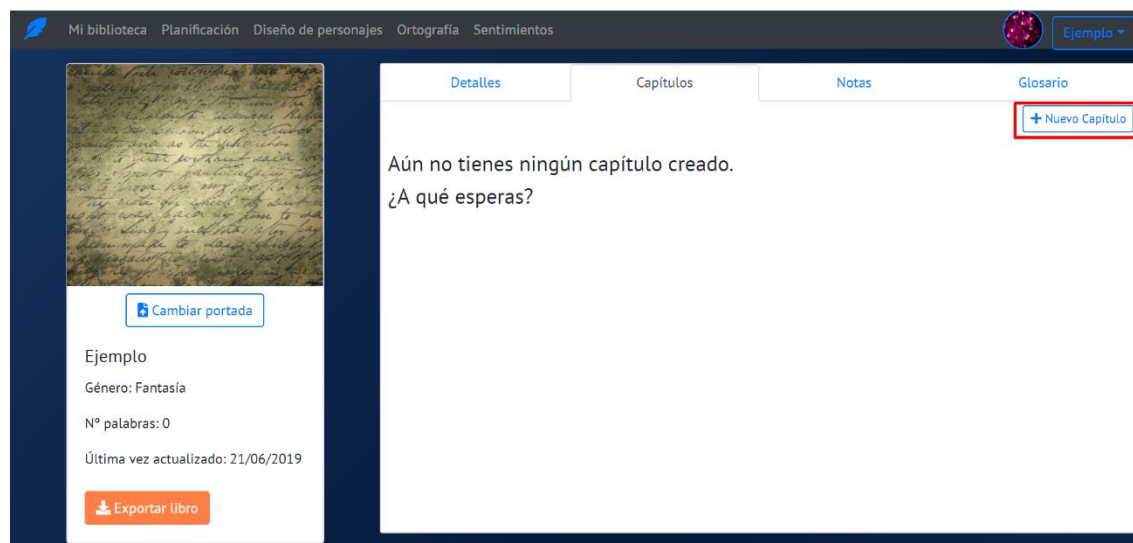


Figura 93 – Página de capítulos de un libro

Al hacerlo redireccionará a la pantalla de escritura en línea.

Otra forma de acceder a ella es pulsando sobre el creado al volver atrás.

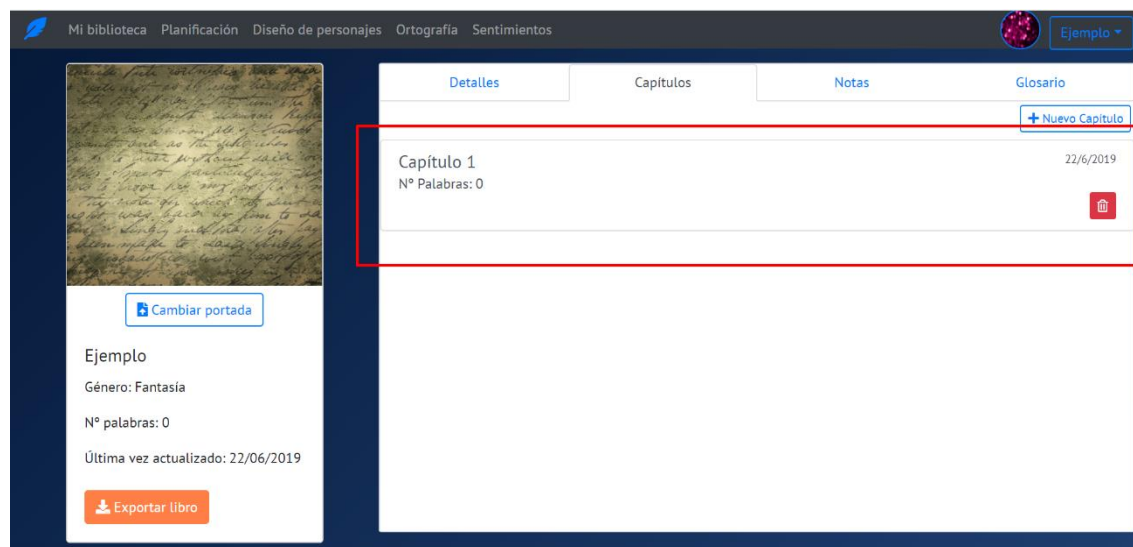


Figura 91 – Acciones sobre los capítulos de un libro

Para borrar el capítulo, pulsar sobre la papelera roja, y confirmar el borrado en la ventana de confirmación.

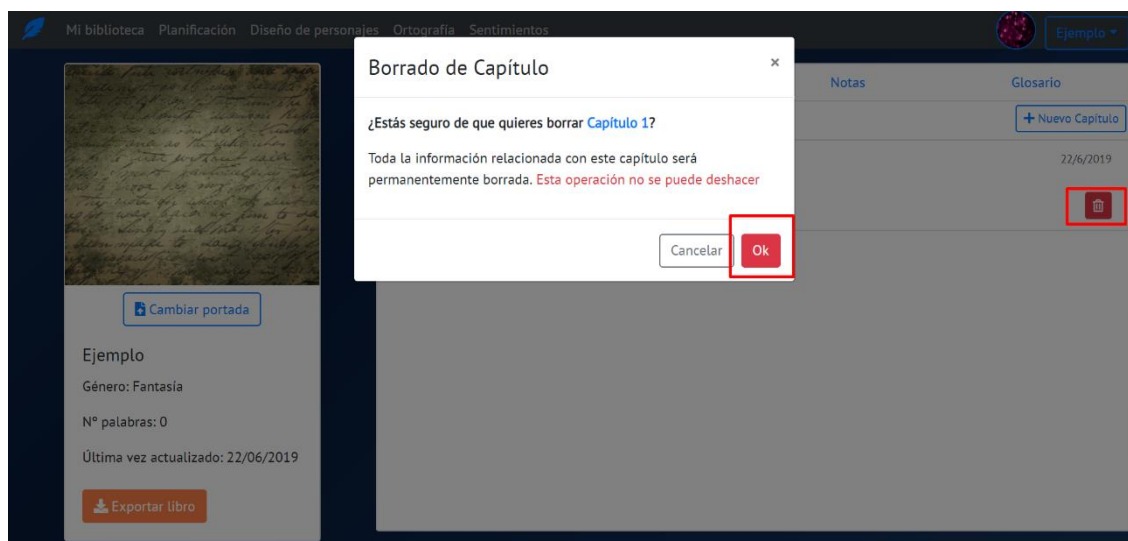


Figura 94 – Página eliminación de capítulo

Escribir Capítulo

Desde la pantalla de escritura en línea, se visualiza el editor de texto. Escribir sobre él el contenido del capítulo.

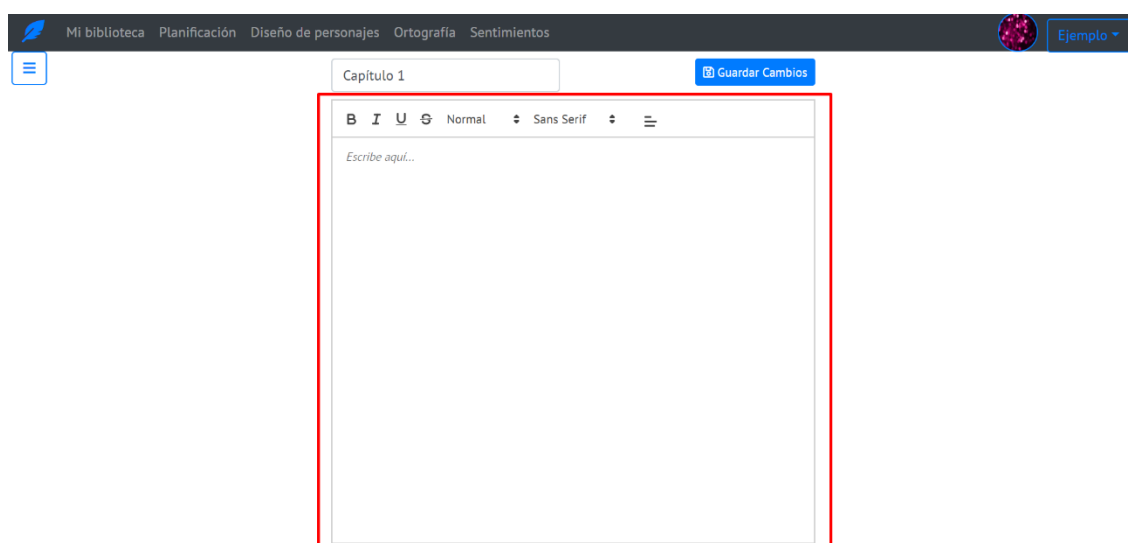


Figura 95 – Página de la escritura en línea

Una vez se tiene contenido escrito, pulsar sobre el botón “Guardar Cambios”.

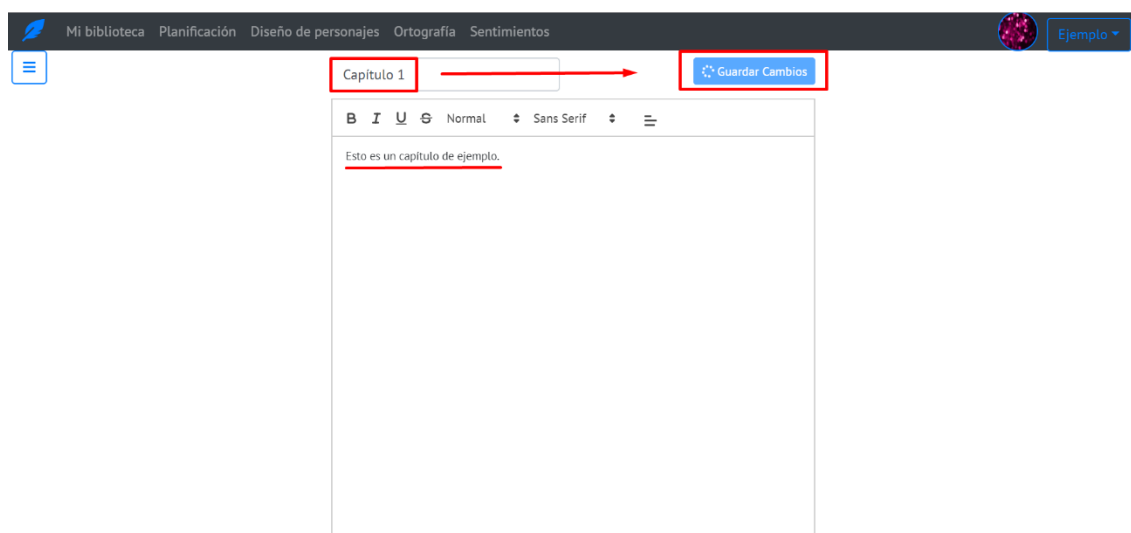


Figura 96 – Acciones en la escritura en línea

Tanto el título del capítulo como el contenido se guardan tras pulsar el botón.

Si se desea cambiar el formato, tamaño o apariencia del contenido escrito, pulsar sobre las opciones en la cabecera del editor.

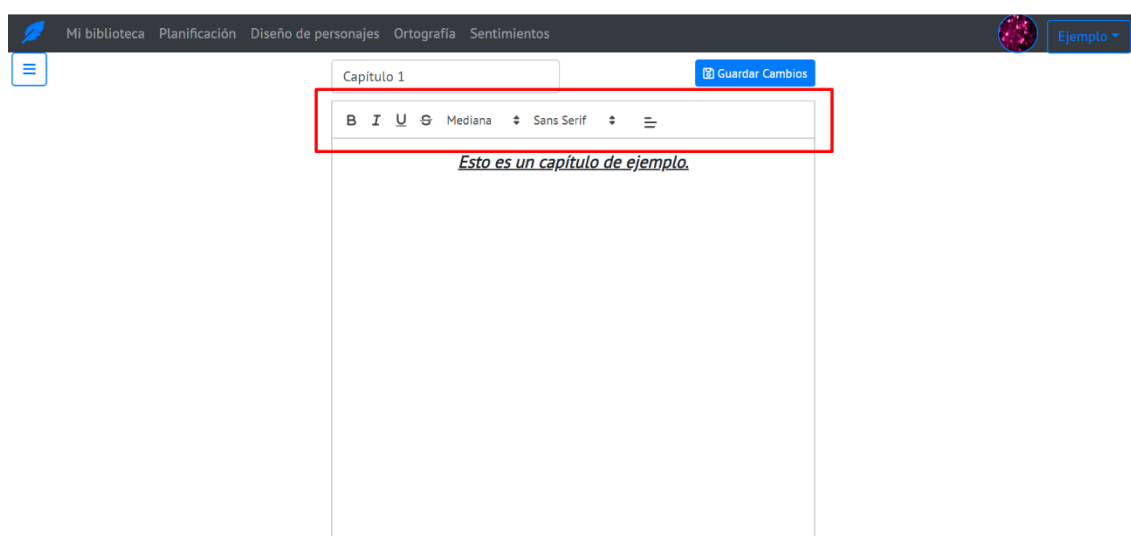


Figura 97 – Acciones de formato de texto en la escritura en línea

Cualquier cambio introducido, se guardará al pulsar sobre el botón de guardado.

Crear Guion Rápido

Desplegar herramientas pulsando el botón de menú desplegable situado arriba a la izquierda de la pantalla.

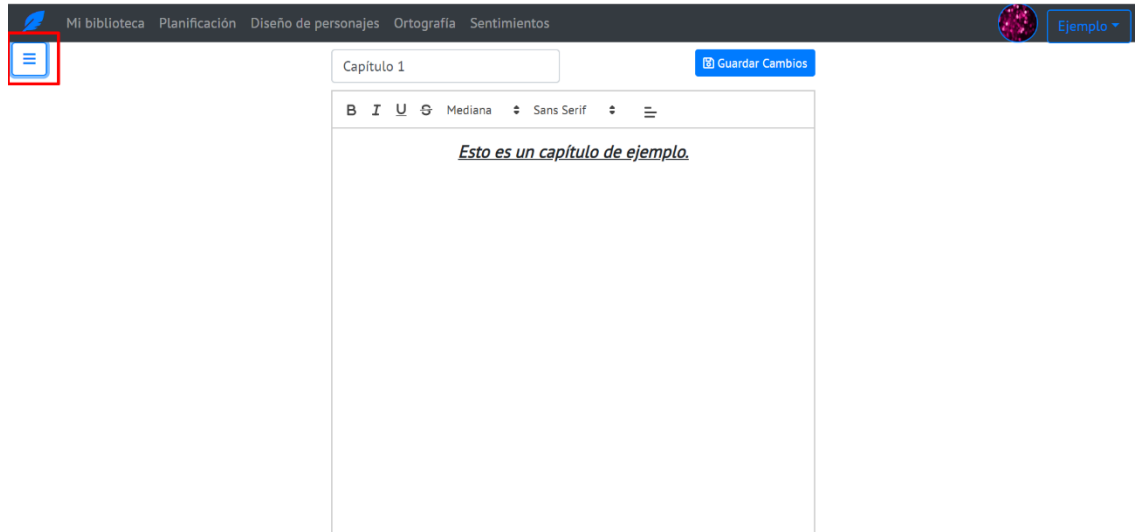


Figura 98 – Escritura en línea activando el guion rápido

Una vez se despliega, introducir el evento a planificar del capítulo y pulsar sobre el botón **+**, a lo que se verá añadido el evento a la lista.

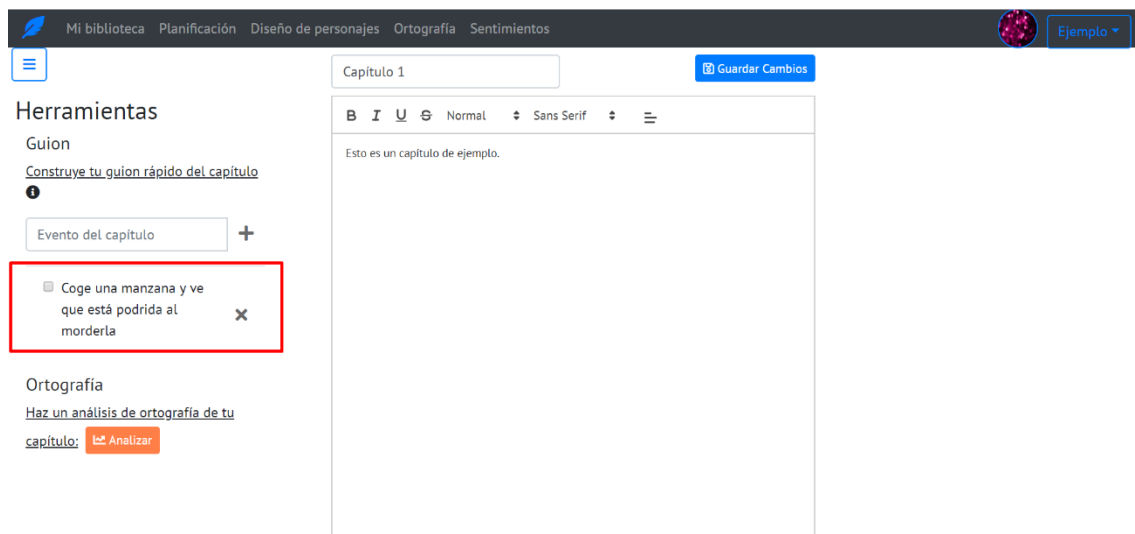


Figura 99 – Guion rápido con elementos

Para marcar el evento como completado pulsar sobre la caja de confirmación de la izquierda del evento.

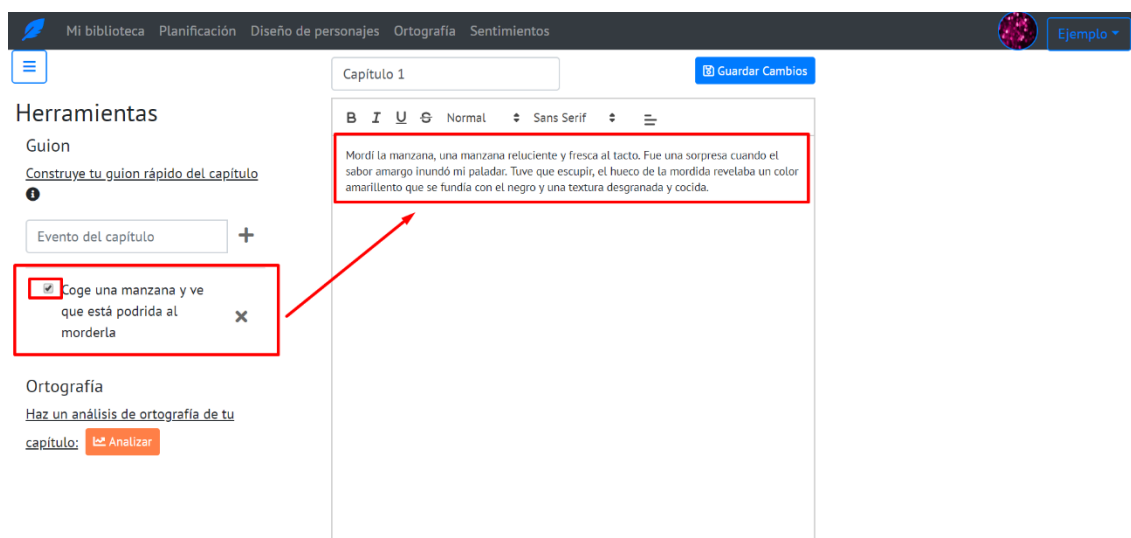


Figura 100 – Evento del guion rápido marcado

Para borrarlo basta con pulsar sobre el botón “X”

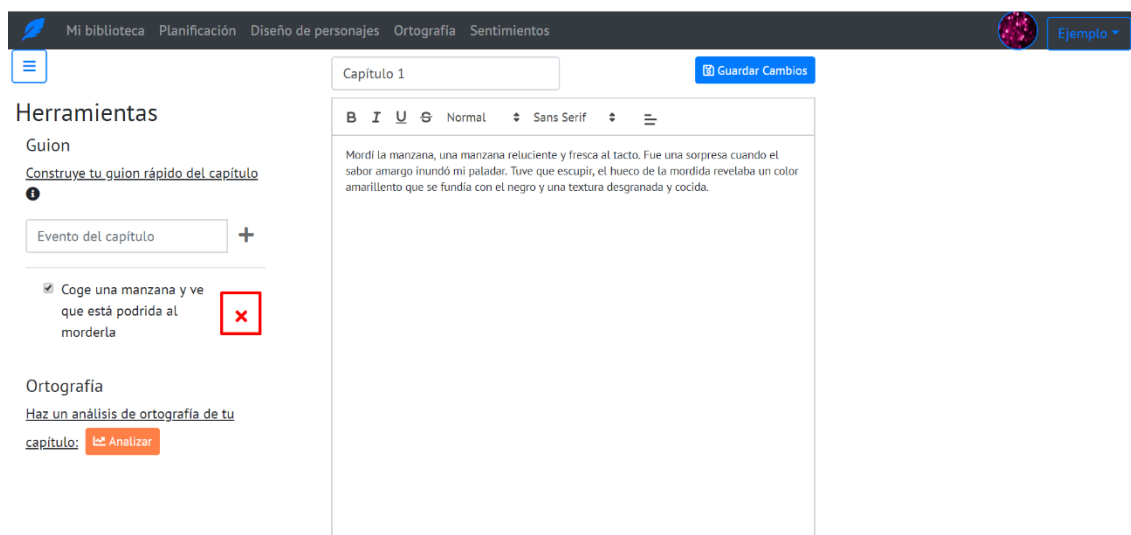


Figura 101 – Eliminar evento del guion rápido

Corrección de ortografía

Con el menú de herramientas desplegado, pulsar sobre el botón “Analizar”, bajo el apartado “Ortografía”.

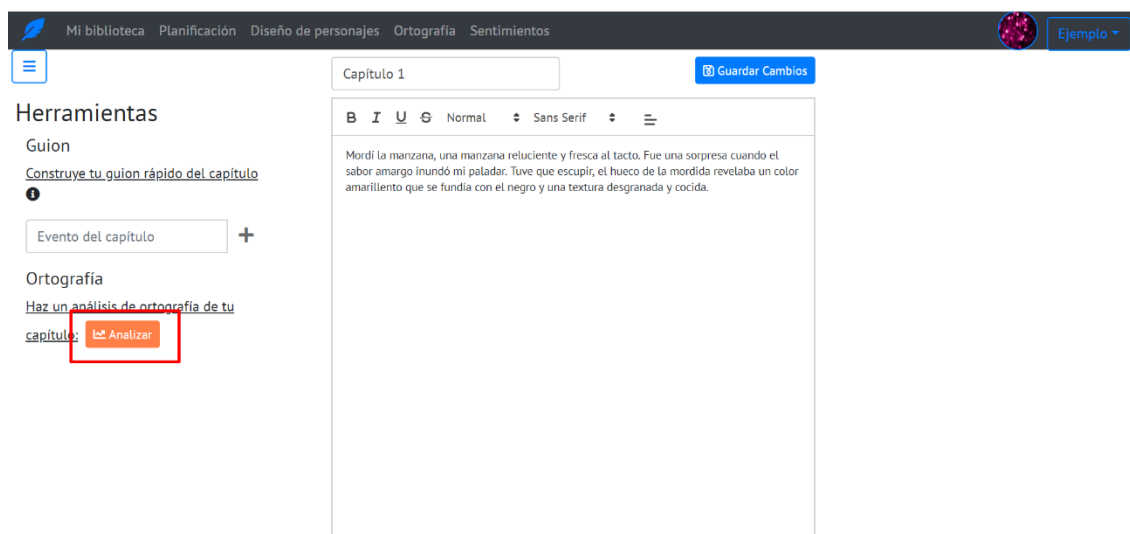


Figura 102 – Acción de análisis ortográfico en escritura en línea

Si detecta faltas, las mostrará en una lista, para ver las sugerencias y corregir la falta, pulsar sobre el elemento de la lista. Después pulsar “Corregir”.

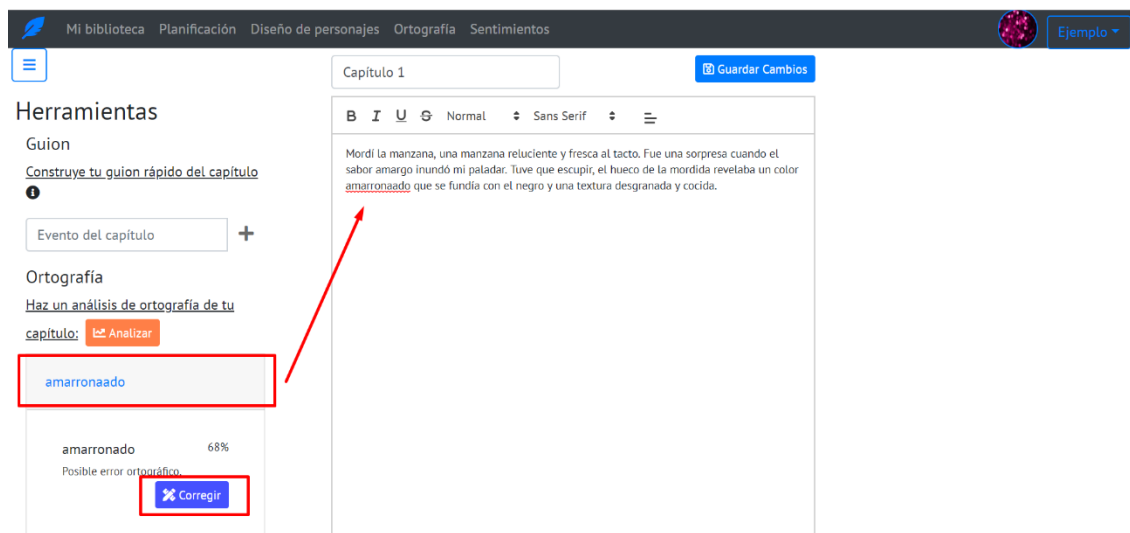


Figura 103 –Análisis ortográfico en escritura en línea

Tras pulsar:

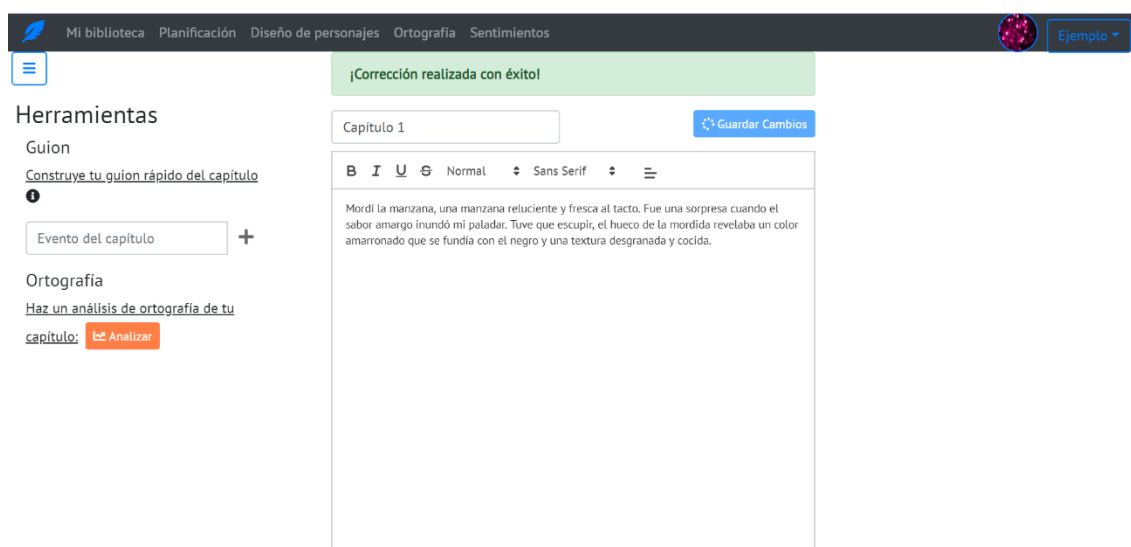


Figura 104 – Error corregido en la escritura en línea

Exportación de libro

Desde la pantalla de detalles del libro, pulsar sobre el botón “Exportar libro” de la carta de presentación.

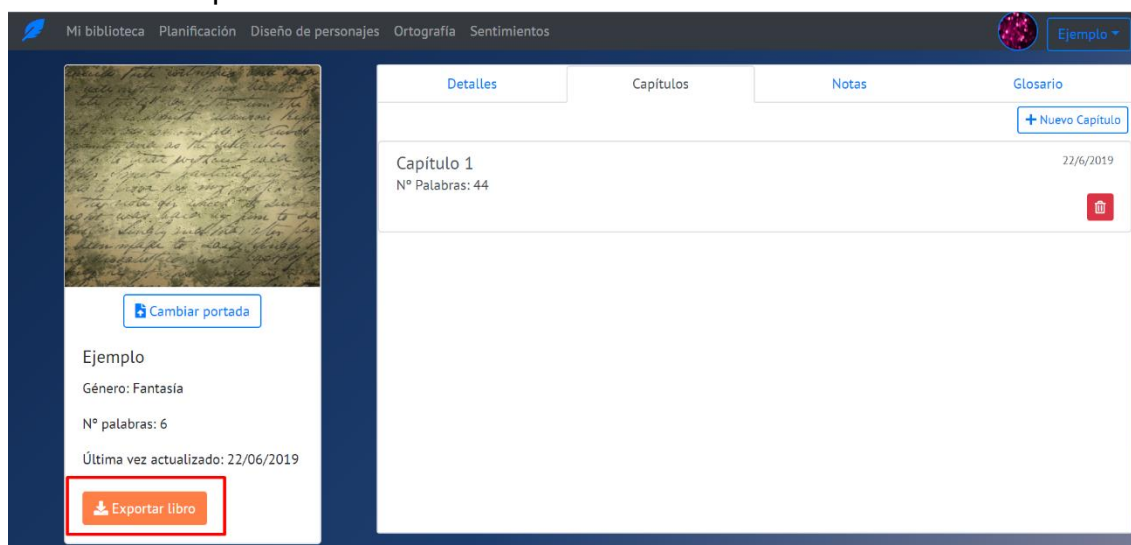


Figura 105 – Acción exportar libro

Se visualizará una descarga, pulsar sobre el documento y visualizar contenido.

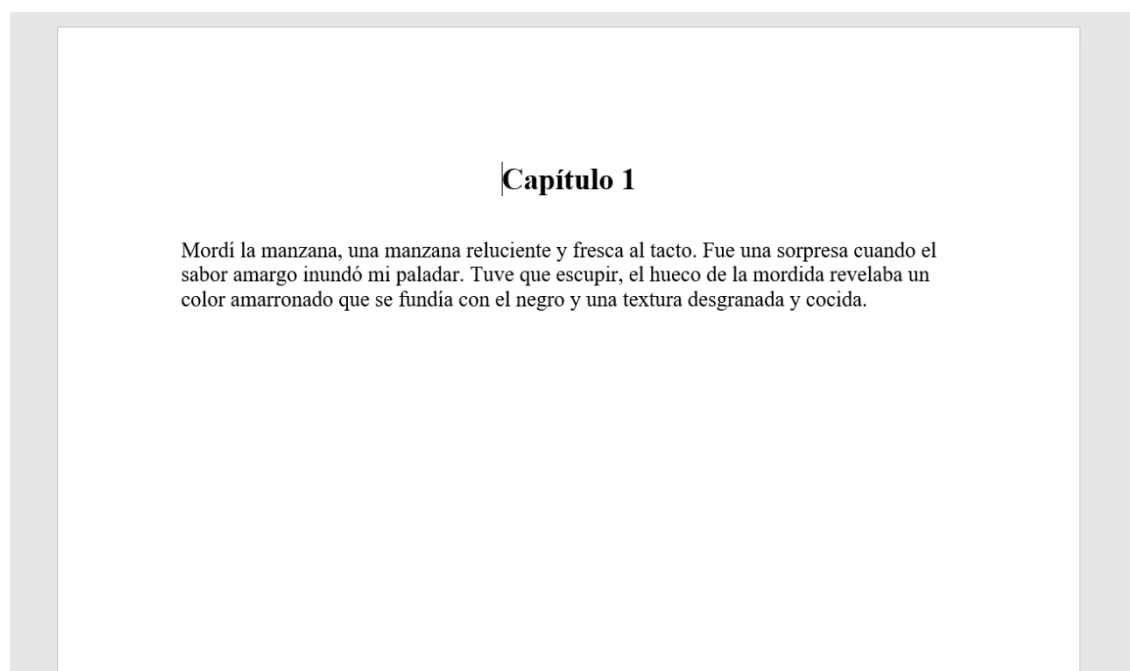


Figura 106 – Libro exportado a formato Word

Gráfica de análisis de ortografía

Pulsar sobre “Ortografía” en la barra de navegación.

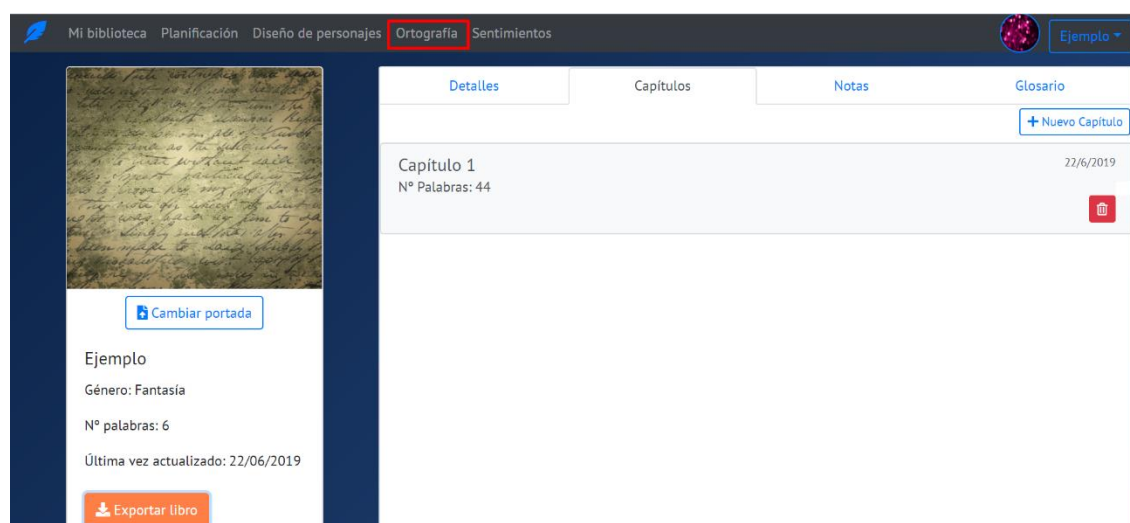


Figura 107 – Navegar a la página de ortografía

Se muestra una pantalla con un texto para escoger libro al que analizar.
Selecciona el libro.



Figura 108 – Selección del libro a buscar faltas de ortografía

Tras ello, pulsar “Analizar”:

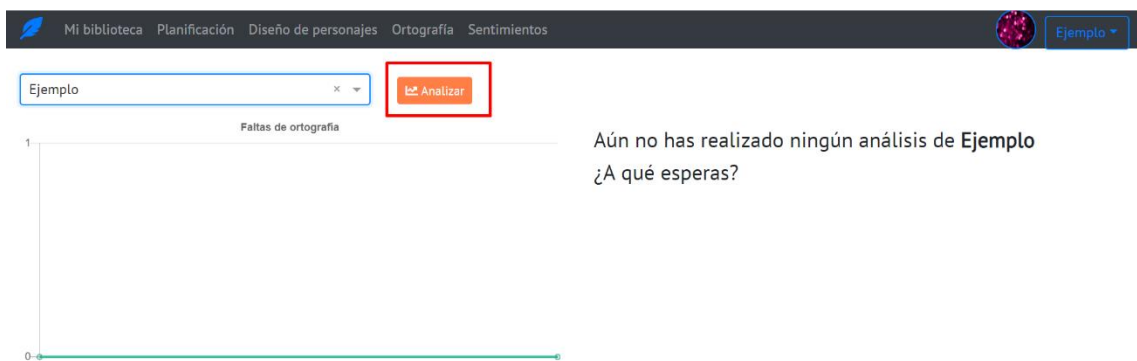


Figura 109 – Acción de análisis ortográfico

Detecta una falta, y a la derecha los capítulos que tienen la errata. Pulsar sobre el capítulo y proceder a corregir pulsando el botón “Corregir”

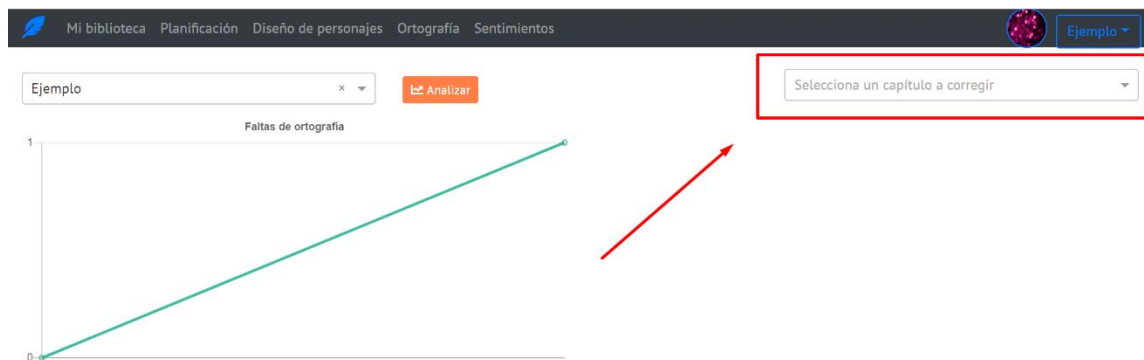


Figura 110 – Acción de selección del capítulo

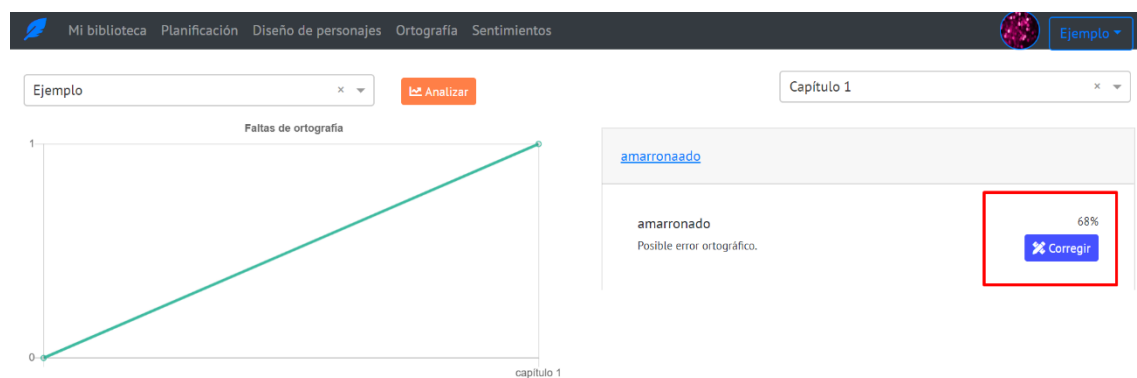


Figura 111 – Acción corregir el error ortográfico

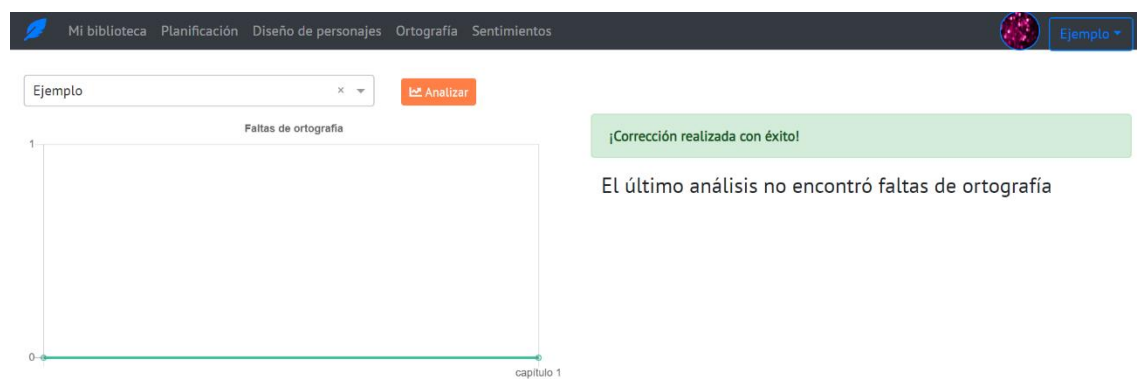


Figura 112 – Mensaje informativo de la corrección de la palabra

Análisis de Sentimientos

Seleccionar “Sentimientos” en la barra de navegación:

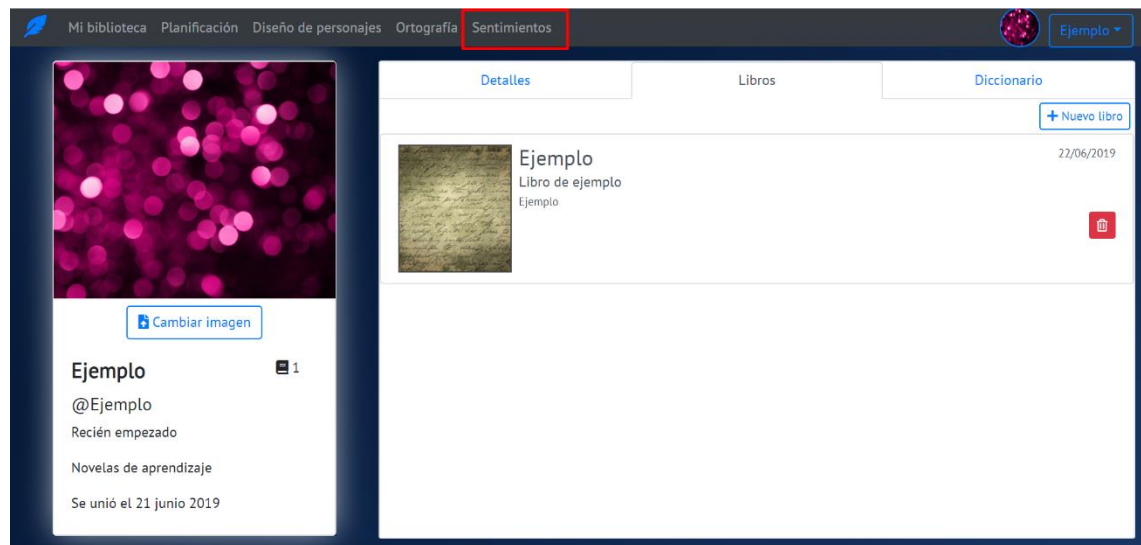


Figura 113 – Acción de navegar a la página de sentimientos

Aparece una pantalla con una gráfica de ejemplo, y un seleccionable con los libros sobre los que hacer análisis. Seleccionar el libro y pulsar “Analizar”:

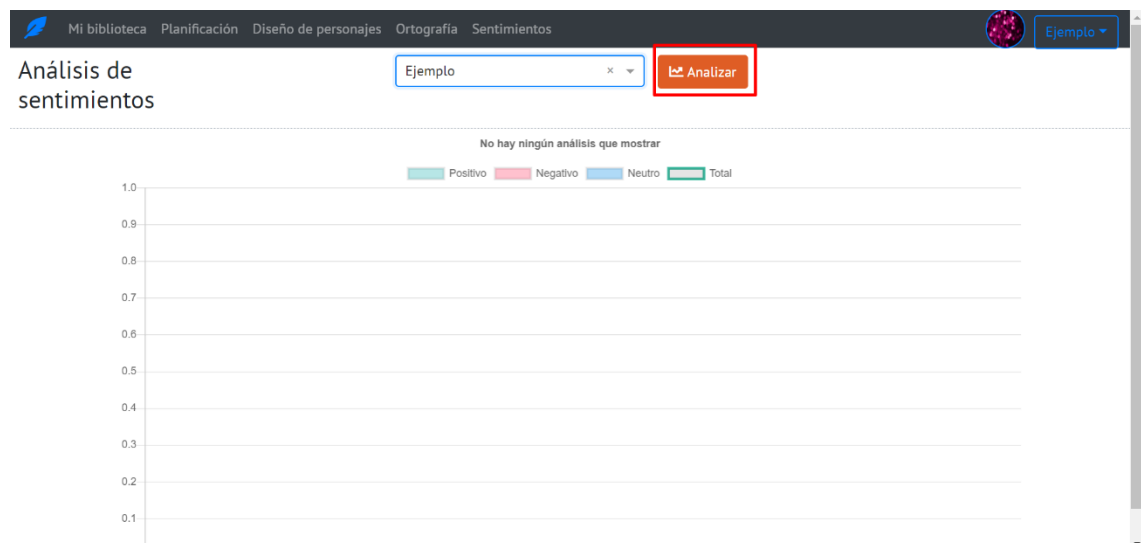


Figura 114 – Acción de análisis de sentimientos al libro seleccionado

Se visualiza el resultado:

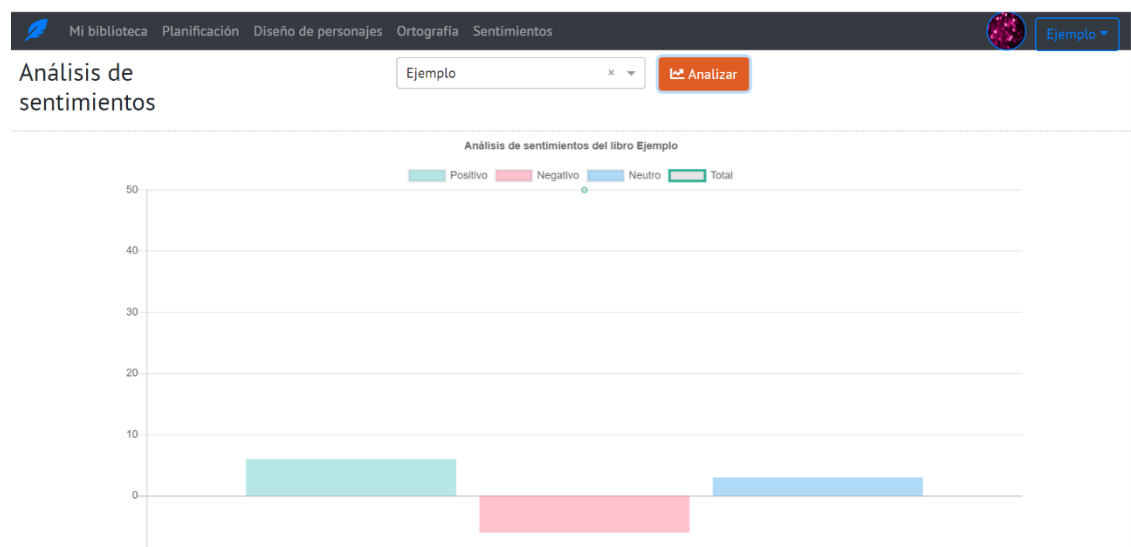


Figura 115 – Resultado del análisis de sentimientos

Diseño de personajes

Seleccionar “Diseño de personajes” en la barra de navegación



Figura 116 – Navegación a la galería de personajes

Se muestran los personajes creados por el autor. Para crear un personaje, pulsar “Añadir personaje”.

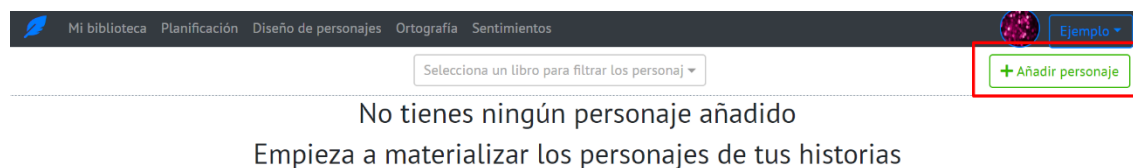


Figura 117 – Acción de crear un nuevo personaje

Se abre un formulario de creación. Para poder guardar el personaje se ha de introducir al menos el nombre y seleccionar el libro al que pertenece.

 This screenshot shows the 'Creación de un nuevo personaje' form. At the top, there is a navigation bar with the same links as Figure 117. Below the navigation bar, the title 'Creación de un nuevo personaje' is followed by a dropdown menu 'Selecciona el libro al que pertenece' and a button 'Elige un libro'. To the right is a green button '+ Guardar'. The form is divided into three tabs: 'Esencial' (selected), 'Aspecto', and 'Personalidad'. The 'Esencial' tab contains a large blue square with a white person icon and a 'Cambiar imagen' button. Below this, there is a list of status messages: 'Sin alias aún', 'Falta definir su orientación', 'Tono de voz no definido', 'Sin rol definido', and 'Última actualización'. To the right of the image area, there are several input fields: 'Nombre*' (highlighted with a red box), 'Alias', 'Rango de edad' (a slider from 0 to 100), 'Género', 'Lugar de nacimiento' (with a placeholder '¿Dónde ha nacido?'), 'Orientación sexual', 'Rol', 'Ancestros', 'Hermanos', 'Hijos', and 'Parejas'. At the bottom of the 'Esencial' tab is a 'Biografía' section with a text area labeled 'Describe más sobre el personaje...'.

Figura 118 – Formulario de creación de personajes, esencial

Se presentan tres pantallas donde incluir información para el personaje, para acceder a cada una de ellas, seleccionar “Esencial”, “Aspecto” o “Personalidad”.

Mi biblioteca Planificación Diseño de personajes Ortografía Sentimientos
 Ejemplo ▾

Creación de un nuevo personaje
 Selecciona el libro al que pertenece ▾
Elige un libro
+ Guardar

Esencial **Aspecto** Personalidad

Compleción
 Compleción del personaje ▾

Tipo de cuerpo
 Tipo de cuerpo ▾

Altura en centímetros (100cm = 1m)
 0 300

Vestimenta
 Tipo de vestimenta ▾

Tono de voz
 Tono de voz ▾

Volumen de voz
 Volumen de voz por defecto ▾

Tono de piel
 Tono de la piel ▾

Tipo de piel
 Tipo de piel ▾

Gesticulación
 Tipo de gesticulación ▾

Vello corporal
 Tipo de velloidad ▾

Color de pelo
 Color del pelo ▾

Tamaño del pelo
 Tamaño del pelo ▾

Color de ojos Tipo de ojos Tipo de nariz Tipo de orejas

Figura 119 – Formulario de creación de personajes, aspecto

Mi biblioteca Planificación Diseño de personajes Ortografía Sentimientos
 Ejemplo ▾

Creación de un nuevo personaje
 Selecciona el libro al que pertenece ▾
Elige un libro
+ Guardar

Esencial Aspecto **Personalidad**

Motivaciones/Objetivos
 Describe las motivaciones y objetivos que tiene el personaje.

Manías
 Describe manías, rarezas y formas únicas que tenga el personaje.

Aficiones
 Describe aficiones, gustos y pasiones que tenga el personaje.

Forma de comportarse
 Describe el comportamiento del personaje en estado natural, ante alguna amenaza, bajo presión...

Creencias/Valores
 Describe las creencias y valores que forjan la personalidad del personaje.

Miedos
 Describe los miedos y temores que tiene.

Figura 120 – Formulario de creación de personajes, personalidad

Pulsar “Guardar” al introducir la información:

The screenshot shows a web application interface for creating a new character. The top navigation bar includes links for 'Mi biblioteca', 'Planificación', 'Diseño de personajes', 'Ortografía', and 'Sentimientos'. The main heading is 'Creación de un nuevo personaje'. A dropdown menu shows 'Ejemplo'. A green button labeled '+ Guardar' is highlighted with a red box. The form is divided into three tabs: 'Esencial', 'Aspecto', and 'Personalidad'. The 'Esencial' tab is active, showing a character profile card with a placeholder image and a 'Cambiar imagen' button. The card lists: 'Ejemplo de personaje', 'Sin alias aún', 'Falta definir su orientación', 'Tono de voz no definido', and 'Sin rol definido'. The 'Personalidad' tab contains various input fields: 'Nombre*' (Ejemplo de personaje), 'Alias' (Alias sobre el personaje), 'Rango de edad' (0 - 0 to 100), 'Género' (Genero del personaje), 'Lugar de nacimiento' (¿Dónde ha nacido?), 'Orientación sexual' (Orientación sexual), 'Rol' (Rol del personaje), 'Ancestros' (Ancestros del personaje), 'Hermanos' (Hermanos del personaje), 'Hijos' (Hijos del personaje), and 'Parejas' (Pareja/s del personaje). A 'Biografía' section with a text area is also present.

Figura 121 – Acción de guardar el personaje

Visualizar el personaje en la galería de personajes:

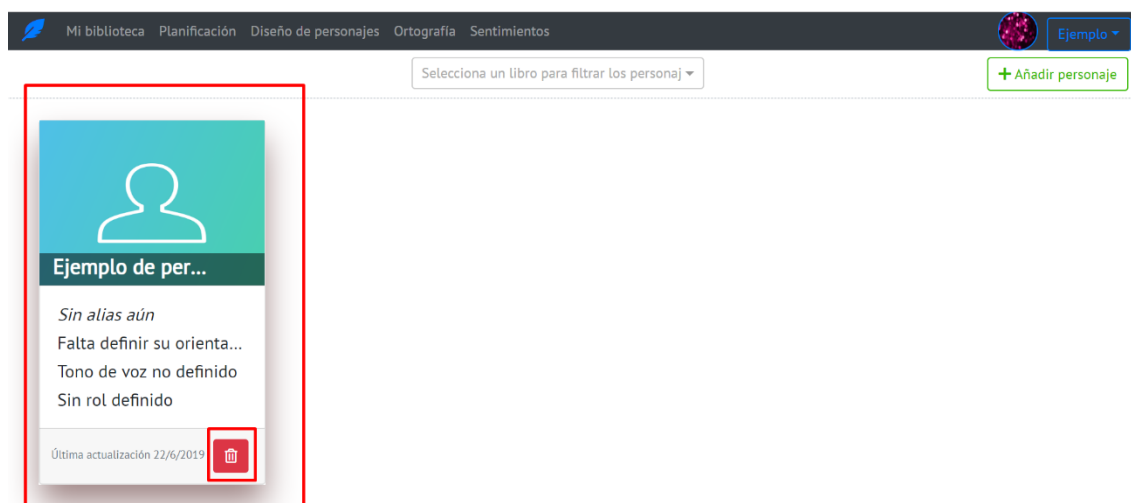


Figura 122 – Acciones de los personajes en la galería

Si se desea borrar, pulsar sobre el botón de la papelera y confirmar en la ventana de confirmación:

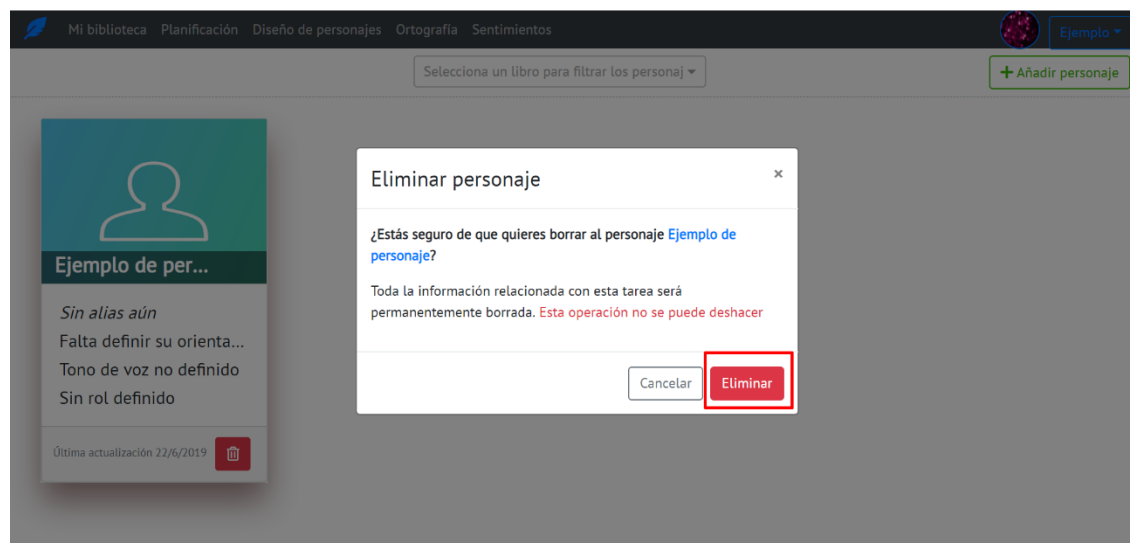


Figura 123 – Acción de eliminar un personaje